[00:01:00] Everybody, welcome to the Friday, September webinar last one of Week 1.
The plan today is we'll have a few one item I promised to go through, which was a numerical application of some properties of the multivariate normal distribution.
That was one of them and I wish to do a quick demo of that using a dataset that we've covered some core challenges.
There was one item.
Then there was a question for Melisa Simpson about actually would turn out to be a typo in the solutions for the Q2 challenge.
I will actually go through that and I'll also show you some other R packages that will simplify doing two-sample Hotelling's T squared testing.
Then there was a question from Young Yung Ho Kim.
If I'm mispronouncing your name, I apologise.
Please correct [00:02:00] me.
That goes back to some of the Week 0 materials about Eigen decomposition and precisely when we write a square root of a matrix, why do we write it that way? I'll pull up the document camera for that.
Now, Wei Kaylao had a question about prediction for under the normal distribution, why do we end up with a linear predictor there? Then, I think Sarah Ray had a last minute question about a certain step in the derivation for the multivariate normal MLE.
Any questions before we start? First things first, [00:03:00] let me share, let me pull up RStudio here and I will do a bit of a demo.
There it is.
RStudio.
Let me clear the workspace just to keep things clean.
This is going to be using the wine dataset which you also saw in your demos and challenges and the goal here and this Help Window is distracting, relates to some other project.
The goal here is to illustrate this idea of conditional distribution under normality.
I recall we have this dataset the winequality dataset and I'm going to read it in now and perform some transformations [00:04:00] that were again suggested in the challenges.
Just to remind you, here's the plot, the pairwise plot of all the variables involved.
Now they're not perfectly normal, but they're not too bad and you can probably make a case for each of the transformation used.
The dataset is called Red T.
Let's split it up into two components.
Now Red T to look at RStudio here.
These are the variables here in the last two are our [00:05:00] log fixed acidity and log volatile acidity variables number 6 and 7.
We'll call them X_1.
Those are the variables we'll be predicting and the rest will be X_2.
Now I don't know whether you've seen this necessarily in our but when you have a minus before the indices it'll basically say take everything but those elements.
So far so good? Let's compute the means of X_1 and the means of X_2.
Here they are.
It might fixed acidity and volatile is 30 and then the means of the rest of them.
Let's compute what would in the notes to be sigma 1,1 and sigma 2,2 and sigma 1,2, which if transported, we get sigma 2,1.
These are the variances of the variables we're predicting variances of variables [00:06:00] and covariances of the variables we're predicting from and then the covariance between the two sets of variables.

Now we implement a formula for prediction.
We start with new one, then we have sigma 1,2, sigma 2,2 inverse and then I'm just going to use M_1 minus M_2.
This is predicting the value of X_1.
The expected value of X-1, given X_2 at whatever the mean of X_2 is.
We'll replace that with something else later.
Then we can run that and get the prediction for the mean.
Now, one thing to notice is that it actually matches [00:07:00] these values, there it is.
It also matches the values for the original mean.
That makes sense because this cancels out, so this whole thing cancels out and so our prediction.
Then for the variance, we have this variance covariance matrix.
Now here we can compare for reference.
This is the variance matrix before conditioning and you can see that the variances are quite a bit smaller.
This is because there is some covariance between the X_1 and X_2, which means that X_2 explains some of the variation in X_1.
Makes sense? Now, [00:08:00] next week we will look at some more formal and more precise ways to quantify and testing how much predictive power X_2 has about X_1.
But this is already a bit of an intuition for you for now.
This is basically how you would use this prediction.
Again if you have a bunch of variables and their multivariate normally distributed then one set of variables given another set of variables will be normal and we know we could be able to derive it, the mean and the variance.
A related idea is what's called multivariate linear models.
[00:09:00] We will get to that in Week 3.
This term is really short.
But for now, let's say X_1, I'll just fit one for you.
I think we should do the trick.
My apologies.
It can be a bit picky.
Really, these should be DataFrame there.
Now we have essentially this coefficients of X_2.
Each element of X_2, predicting each of the elements of X_1.
Now look at these coefficients and look at this expression.
[00:10:00] Yes, good question.
The formulas in the notes one second I actually have the time here.
There it is.
These are the properties of the multivariate normal.
We're looking at an application of property number 4.
No worries.
But notice that for example, this 0.
54 log fixed [00:11:00] acidity versus density happens to equal to this one, and this one happens to be equal to this one, and so on.
That are here on coincidences.
There's a very close relationship between multivariate normality and linear model.
In particular, multivariate normality is the scenario under which linear modelling is the optimal prediction method.
In fact, I think this should match, although might not match exactly.

Again, we'll cover these in more detail in Week 3, but the numbers aren't exactly the same because there is a n minus 1 factor for [00:12:00] some other, for others.
But the idea is that this is the variability in x leftover after we've taken out all of X2.
This is the numerical demonstration of that particular property of the normal distribution.
Questions? Well, actually I have a question for you.
You can see that there is an intercept effect here as well.
This is linear regression.
How do you think can we compute this intercept from these values? [00:13:00] I'll give you a hint.
Almost.
We're predicting X1 given X2 so actually, let's set X2 to 0.
In fact, if we set this to 0, and if we just evaluate, I'm going to cut a little bit sloppy here and just paste it in down here.
If we evaluate this at 0, which is just minus M2, we get minus 50.
0 and 0.
6 minus 30.
9, so the numbers up here.
Now we could plug in other [00:14:00] values.
For example, we have, so let's see, we have U2.
Well, maybe I should have picked a shorter vector.
But let's, I'm just going to pick some arbitrary numbers here.
I don't know, 1, 2 minus 1, 2, 2.
If I ran that, these are now the predicted log fixed acidity in long row total acidity for that combination of levels of density, pH, log sulphates free sulphur dioxide, and sulphur dioxide appropriately transformed.
Any questions? [00:15:00] Now, on further thought, I think what I will do is I will do YK allows questions and each question next because it did actually concerns the same slide so I'll just illustrate the prediction.
Now one thing to observe is that this works just as well when you have only one dimension.
If X1 only has one dimension and the rest is X2, that works just fine.
You just get a scalar predictor.
The next question was again by YK Lao.
I'll get them doing them a little bit out of order.
It also concerns properties of the multivariate normal.
[00:16:00] For this question, I will turn around the background camera.
The question was, how do we go from this? How do we get [00:17:00] this result from the regression function is the prediction for Y hat as a function of x, is under multivariate normal, and in this case, y is a scalar, but actually the vector is the expected value of Y plus Sigma naught transpose C inverse the expected value of X vector.
Now we're here.
[00:18:00] C is defined as the variance of that covariance matrix of X and Sigma naught.
I'm sorry, this is copied from wrong place, sorry.
In my notes, X this is of course x minus the expected value of x.
This is covariance between y and x.
Well, having just seen the demonstration before, you can probably recognise some parts of this expression.
If we use the notation from that slide.
Again, unfortunately, I can't show both the document camera in the slide.
But you recall [00:19:00] that in the slide we had a split between X1 and X2.

Both vectors.
Well now we're going to say that this one is, we'll call this one Y and we'll call this one just X.
Here, all of a sudden this covariance becomes Sigma 12, and this becomes Mu1, and this becomes Mu 2.
This of course becomes Sigma 2.
If we just plug these expressions, it actually simplifies to this regression function.
[00:20:00] Which again relates to this idea of which one did you get to relate to this idea of normality to a new regression? Does it make sense? YK if you are there? That's that.
Now that slide also has something very useful for one of the assignment questions and explains what does it mean to minimise the squared error of Y given X? I'd recommend taking a look at that as well.
[00:21:00] Starts that.
Now I think next item was I wanted to go through.
Sorry.
Which part? The part after Property 5.
Yes.
Let me see.
I'll do it after I go through the planned stuff.
Is that all right? [00:22:00] It's easier to keep track because I want to post timestamps after, so I want to keep things organised.
Then the next item.
The question was from Young Ho Kim.
Again, I apologise I'm mispronouncing it.
This one concerns the question of computing the matrix square root.
This basically concerns the slide on matrix algebra.
In particular the eigenvalues and eigenvectors and the equation 0.
10.
Again, rather than switch between cameras, I will [00:23:00] just reproduce it here.
The expression was this.
If we had an eigen decomposition X, which is in this case a matrix equals sum and X is p by p here of the eigenvalue times the i_eigenvector times i_eigenvector transpose, and so the question was, how is it that we can use this to get, take a square root of a matrix.
Well, we define the square root of a matrix if it's in that context as [00:24:00] I equals 1^p of Lambda i to the 1/2 and then again e_i, e_i transpose.
Now, why is this the square root? Well, it turns out to be pretty straightforward.
Considering the properties is we can write this out as a summation.
X^1/2 equals 1.
We can write it down as a summation.
But actually now that I think about it, this is actually more cumbersome than the other approach, which is [00:25:00] to remember we're writing this as P Lambda P, where P is a matrix with these vectors as columns and Lambda is a diagonal matrix with Lambda as columns, sorry, as elements on the diagonal.
You transpose, so then that means that, actually I'm going to draw this out.
We have P then Lambda to the 1/2 power.
Now, we can't just take any matrix to the 1/2 power.
I'm just going to write this out.
This is Lambda to the 1/2, Lambda 1 rather to the 1/2, all the way to Lambda P to the 1/2.
Then you have P transpose, and then we have the second X to the 1/2, which is P.
Then again repeat this matrix Lambda to the 1/2, [00:26:00] Lambda P to the 1/2.
Then again P transpose.

Now what happens? Well, we have P transpose P here and P is orthogonal.
This was just identity matrix.
It goes away.
Now we have P times Lambda to the 1/2 matrix times one from the 1/2 matrix with nothing off the diagonals.
But the way multiplication works for diagonal matrices is we just get P times Lambda 1^1/2 and then squared, Lambda 1^1/2 squared, and then P transpose.
But that just gives us P Lambda P transpose, which is the [00:27:00] original X.
That shows that this is in fact a matrix square root.
Now, I could also write out this matrix multiplication in terms of summation.
Well, in terms of these summations, it's a bit cumbersome.
In particular, we would write it something like, so we have the sum i equals 1^p Lambda i^1/2 e_i, e_i transposed.
Then again, sum, this time, say call it j equals 1^p of Lambda to the 1/2 e_j [00:28:00] e_j transpose.
Well, the way we could do this if we wanted to would be that we would pull out the two summations, so i equals 1^p, j equals 1^p.
Then we would have Lambda i^1/2, lambda j to the 1/2.
Then we would have this product.
We have e_i, e_i transposed e_j, e_j transposed.
[00:29:00] Here's the thing.
Yes, this is correct.
Now, here's what would happen.
Well, can anybody suggests what would actually happen here? Like what can we simplify? Here's a hint.
Remember that these e_i are orthogonal to each other.
[00:30:00] That's a start.
But first of all, that's not quite right because each of these is a column vector.
Transpose column vector actually creates a matrix.
A column vector times a row vector creates a matrix.
But you're on the right track.
They detect e_i transpose e_j, which is the dot product between the two.
That is zero [00:31:00] if i does not equal to j.
But right here, we have precisely this expression.
We have e_i transpose e_j.
Now, that's going to be zero unless i equals j, which lets us simplify this expression.
Now we only care about the case where i equals j, so we can write this as Lambda i^1/2.
Lambda i again to the 1/2.
Then you have e_i, e_i transpose, e_i, e_i transpose all the vectors.
Now what can we do now? [00:32:00] We still have an e_i, e_i transpose.
But now it's the same vector and we know that that equals to 1, again orthogonality.
Now what we have, and here, by the way, we have, of course Lambda^1/2, so we can just get rid of the 1/2.
Now we have the sum of Lambda i, i_e, e_i transpose, which is just back to x.
These are different ways of showing that in fact, this equation does give us a square root of a matrix.
[00:33:00] The next question was by Sarah Ray.
I think just wanted to put one in.
I'm trying to do all the document camera ones at once.
This question was this.
We had this expression in the derivation of the MLE for the mean, which is that you have trace of Sigma

inverse.
Then you have the sum of x_i minus x bar, [00:34:00] x_i minus x bar transpose.
I'm going to add the vector bits.
This is Sigma inverse.
This should make the Sigma different from summation somehow.
I equals 1 to n there.
Now it's clear.
Transpose T plus n, x-bar minus Mu, x bar minus Mu transpose, and then close parenthesis.
This trace, we claim equal to 2, the trace of Sigma inverse [00:35:00] times this summation.
Just for brevity, I'm going to take this and label it sum of squared residuals, so SSR.
Then plus n x bar minus Mu Sigma transpose Sigma inverse x bar minus Mu.
By the way, these are all vectors, x_i, x-bar, Mu, n is [00:36:00] not a vector, and Sigma is a matrix.
The question is, how did we go from here to here? The answer is, we use the property of the trace.
What we actually did is that we took this expression and we can expand it.
We can say that this is Sigma inverse and then SSR, sum of squared residuals.
Then plus Sigma inverse n and then x bar minus Mu, x bar minus Mu transpose.
Then everything is under the trace.
Now, the trace, you recall is the sum of the diagonal elements of the matrix.
[00:37:00] It's pretty obvious to include the trace of a sum as the sum of the traces.
I think it's in note somewhere.
We have trace of Sigma inverse SSR plus the trace of Sigma inverse, and then basically everything in here.
One property of the trace that we know about is that, if you have a matrix product inside a trace, you can permute the matrices, or at least you can switch the order [00:38:00] of multiplication.
You have to be careful.
You have to do it one at a time.
You can't just permute them however you want, and it has to be conformable.
But what we can do is we could take this and that's as a scale.
I've put it over here.
Then we can take this x minus Mu transpose.
Again, because we're inside the trace, we can move it over here as well in front of the Sigma inverse.
That will give us this expression.
That will put the Sigma inverse in-between x minus Mu transpose and x minus Mu.
Does that make sense? That [00:39:00] you can permute matrix multiplication under trace is one of those properties that you wouldn't think it would hold, but it does.
It's one of those surprising things in my opinion.
Any questions? The next question was, there after we reshuffled things a bit.
Luisa Simpson pointed out an error in the t-squared challenge and so I'm going to go back to screen-sharing and first of all clarify there, but also talk about some other things around that.
[00:40:00] Any questions while I'm still on document camera? What you should be seeing here is a bit different from what you are seeing here, which is annoying there.
I think we are good.
These are solutions I updated a few hours ago based on, again, thank you for reporting the error.
Again, we write it in a dataset, [00:41:00] we use the transformation suggested last week.
Then the tasks were mostly about plotting the data and 95 percent ellipsoid for log fixed acidity in water, there it was.
Then there were some exercises around confidence intervals which may be useful in the assignment.
Then there were two ways to compute confidence intervals.

There was the contrast CI method here using the F distribution and then there was the Bonferroni CI which uses the T-distribution but it uses a different confidence level.
Now, [00:42:00] the pros should be bit familiar since we talked about the concepts between those at the last webinar.
Now here there was an interesting example here.
The was interesting example here about comparing Bonferroni and these ellipsoid confidence intervals for all seven variables in the dataset.
We actually see that before we use the def function to look at the difference between the lower bound and the upper bound of the confidence interval and what we see [00:43:00] here is that actually the Bonferroni ones are narrower and the reason they're narrower is because again the F distribution CIs, they incorporate every possible linear combination.
That's like reporting the confidence intervals for things like, all the sums and differences between these variables.
Then there were the confidence intervals and test for t.
Here I asked you to try to implement these for yourself.
Again, this is my implementation.
We compute the column means, the covariances for the two groups, and then we compute the pooled covariance matrix, and that's pretty straightforward.
We [00:44:00] take the covariance matrix for each individual group and then we multiply it by the sample size for that group minus 1 divided by the total sample size minus 2 that's pooled.
Then here we have the formula for the test statistic and critical value and then the p-value which is again we compute the F statistic and get the p-value.
Now, one thing I just started to this challenge is that there are packages that implement this test already.
For example the package ICSNP has a Hotelling's T2 function which does exactly what you'd expect.
Now here you have to be careful because the statistical reports can be different.
In particular, there's a t squared.
You can sound a report [00:45:00] to the t-squared statistic, others report of the equivalent F statistic and unfortunately, it looks like I didn't print a value here, but it should match whatever is here.
Our curve is not really about t-tests, but it does have the t-test component to it and here it is.
You can see the t-squared matches and it reports the corresponding F statistic as well here which is reported for this function as the F statistic, which is a bit annoying.
Then there's ergm package which actually I maintain and actually wrote this function a long time ago to do something completely unrelated.
But you can do a Hotelling's two-sample plus t-squared test for this as well.
[00:46:00] Var equal means you pool it and assuming that's because this function actually is supposed to handle correlated data as well and this is saying assuming it's uncorrelated.
Anyway, same test statistic, t squared, and same p-value.
Now for unequal variances, the test statistic now we have the variance is taken individually like so, so you now have a somewhat different t-squared.
Here you define the trace function as the sum of the diagonal elements of a matrix.
Then I plug in the ridiculous formula for nu that you saw in the notes.
Again, we're starting to get into where it comes from.
[00:47:00] But this is the degrees of freedom.
Again, the p-value is zero, so it's not.
It's weird, but as far as I can tell, on crown, the only implementation of Hotelling's T-squared test with unequal variances is actually my package.
I don't know why, but I don't think nobody else voluntary implemented.

But here's implementation and here's the call.
You just saved var.
equal equals FALSE.
We still assume independence because this is an independent sample that is uncorrelated and so there you go.
That's what I have for that particular item.
Any questions? [00:48:00] Then I think the last item was going through the multivariate normal distribution properties in particular, the discussion covering prediction here.
Let's talk through it.
We have a variable y, it's a scalar and there are p predictors, X_1 through X_p.
Now, there are different ways we can decide what's the best prediction [00:49:00] and one of those is to minimise the mean squared error.
Let's take it bit by bit here, its expected value over the possible values of Y of Y minus our prediction g of X and then the quantity squared for a given X.
We can choose whatever functions g of X we want provided that its square is finite.
One second.
Ben? Can I just ask probably a stupid question.
Are we looking for one of those X_i that is the best predictor of Y of some new combination [inaudible 00:49:57]? [00:50:00] Right now, g of X is allowed to be any function of all the Xs.
It can be just one of them.
Maybe we don't need the others or it could be some linear combination with them or could be some non-linear combination of them for the moment.
Okay.
What the next argument shows is that, we can actually use the normal calculus to answer this question because for a given X well, if we take a derivative of this, then under fairly common regularity conditions, we can interchange this expectation with differentiation because expectation is a sum of a derivative or a sum or an integral with respect to different integrand with respect to a different [00:51:00] variable than the one we are differentiating with respect to.
But now it turns out to just be that our derivative is minus 2 the expected value with respect to Y of a Y minus g star of X, which we then set to 0.
Well, the only way that can be the case universally is if we set the g star of X to be equal to the expected value of Y given X.
Now, in some sense, the choice of the square difference here is a bit arbitrary.
We could have used something else.
For example, we could have used the absolute value instead.
[00:52:00] It turns out that in that case further than the expected value that is the population mean of Y given X, it would have been the median of Y given X.
That's a separate discussion.
In fact, you can pick all different loss functions here, but that's again, a bit outside the scope of this discussion.
This goes back to the demonstration earlier to start with the RStudio.
Well, for normal distribution, we know exactly what the expected value of Y given X is.
We know we can get it from Property Number 4.
It's this bit really.
When we plug in that and then simplify here, we get an expression that looks like this.
That's how all that [00:53:00] ties together.
Make sense? That makes sense.
If I could ask about its relation to one, say in the quiz, it feels to me like C and D look like the same thing

to me.
I'm happy to clarify.
They're not the same.
Let me pull up my preview of the quiz.
What I see might tiny a bit different from what you see.
For example, I get the option to edit the questions.
You don't.
At least I hope not, because if you do then there might be issues with the website [00:54:00] configuration.
But there's CMD.
Here we have the best approximation in the sense described in those slides.
This is basically the definition of the problem that we had before.
We know that the function x_2 that satisfies this criterion is in fact the expected value of X_1 given X_2.
That's something that the result shows.
But now the next part asks, fine, you found a function that does this minimisation, now, tell me what this mean square error actually is.
You computed this expected value.
Now [00:55:00] find me the expected squared difference between X_1 and your prediction for a given X_2.
That's the part there.
What does your minimum turn out to be? Does that make sense, Ben? Yeah, that makes lot of sense.
Thank you.
.
Mr.
Odin.
Again, if I mispronounced your name, I apologise.
It needs to be an expression in terms of [00:56:00] X_1, X_2, and basic operations.
Any other questions? That's all the questions on the forum at least until a few minutes before the webinar.
Nasim? Well, I think I pretty much [00:57:00] explained it.
You compute the expected value, but here you have to compute the expected value of a somewhat different function.
You have to plug in the g* that you figured out in part C and take the integral, whatever it is, of this expression? Questions.
Go ahead.
Another quiz question.
When we're looking at the confidence regions for the ellipses.
Part B asks us about the known population covariance matrix [inaudible 00:57:51].
Can you maybe point me to where I should be looking at in the notes [00:58:00] for that because I can say exactly where it is, but I'm bit stuck with.
This one is a bit challenging, but should give you a hint.
Let me show where it is.
Question, where did I put that a bit? There it is.
The hint is we talk about the known covariance matrix here.
The question is, what can we do here [00:59:00] as opposed to in the unknown case, which is more complicated, so we've covered in a separate slide? That's the hint.
Thanks.
I didn't notice that.

I put together I will discuss the solutions after they're due.
Well, depending on whether there are any, and you could even considerations what we'll do probably do it.
Well, I think Monday may be a bit premature because there might be some issues submitting for some people with some [01:00:00] students, but maybe next Wednesday I will discuss the solutions.
Any other questions? I think what I'll do is I'll take a 10-minute break and then when I come back, I'll just have consultation, so if you want to ask a question, I'll be watching.
Nasim, actually, can you wait after the break? YK, if you have questions over the weekend, you can ask them on the forum.
I won't promise I'll answer them.
As usual, I might [01:01:00] defer them to the webinar on Monday.
Is it possible [inaudible 01:01:14]? I'm pretty sure that that dataset you can actually download the whole thing.
One second let me find the bit where you can.
I think dataset on RMD file slide.
I think the RMD files might be a little out of date since I made some corrections, but I think the datasets are all there.
[01:02:00] All right, I'm going to take a break.
[01:03:00] [01:04:00] [01:05:00] [01:06:00] [01:07:00] [01:08:00] [01:09:00] [01:10:00] [01:11:00] [01:12:00] [01:13:00] Everybody, welcome back.
I think somebody had their hand up.
Nasim, do you have a question? Nasim? [01:14:00] Well, if you should come back and if you ask a question, I will answer it.
Open? [01:15:00] [01:16:00] [01:17:00] [01:18:00] [01:19:00] Yes, thanks for that.
I'm sorry if you've already answered this somehow in another webinar, but in terms of the answer and uploading the working at the end, I have two questions.
One [01:20:00] is, are there any limits to what we can use for calculating eigenvectors and eigenvalues.
No.
No limits.
Do we have to upload that something that, should we be, is there a mechanism for uploading the text of scripts or something for where are working in those cases, otherwise I'll have no working or I can [inaudible 01:20:25].
[inaudible 01:20:30] write-up that you upload to Turnitin.
I can't see, can you embed the file? Well, you can copy, you can just copy paste them in.
Sure, so is the right up at the end, I was imagining it was scans of my working, but we can also embed other stuff we can do.
Well, actually I would recommend typing it up because I actually turned it in and doesn't like scans unless it's just a figure.
[01:21:00] In fact, I think Turnitin will complain if you don't have at least 20 words that it can recognise? Yeah.
No.
I think this is the second thing somebody asked, so I think I'll make an announcement.
This could be an issue.
If we have a lot of the integration calculations, we have to use latex or something? I don't necessarily want you to.
I'll put it this way, it doesn't have to be written up nicely.
As long as it's reasonably clear that you used some other symbol, for example, instead of an improper integral, then as long as there's a basic idea to any questions about that it's reasonably clear that you

[01:22:00] did do something that is different from what other people have done.
Is it a PDF or something that we upload at the end? PDF, Word document.
I think I should probably make an announcement about this because I don't know whether the actual quiz Moodle site makes it very clear.
Let me try to fix that.
Then I might just embed some of my working as scans within the PDF and there'll plenty of other words in there but as an image and see how it will go with that.
In the end as long as Turnitin doesn't complain, it'll be fine for now.
Thanks.
[01:23:00] I'm sorry.
The people who wrote these instructions don't quite understand how Turnitin works, and I didn't do a manual inspection of that.
Actually, I owe you an apology for that.
I guess, I'll fix that.
For this quiz, I guess, we'll just have to accept whatever you are able to upload; for the next quiz, we'll have to correct that.
I'll post something to that end.
Thank you.
[01:24:00] Sorry, I didn't see your question.
There's no limit as such.
What do you mean? What limit are you asking about? [01:25:00] [01:26:00] I know what you're referring to but are you asking about, say, the file format or file size or when to find that? [01:27:00] Now, this time I will allow handwritten solutions without limit, and then you just have to wait for a sec, I'll be right back.
[01:28:00] [01:29:00] I'm back.
[01:30:00] [01:31:00] [01:32:00] [01:33:00] [01:34:00] [01:35:00] [01:36:00] [01:37:00] [01:38:00] [01:39:00] [01:40:00] [01:41:00] [01:42:00] [01:43:00] We'll be finishing up in five minutes.
[01:44:00] [01:45:00] [01:46:00] Sorry.
One last question.
Yeah.
It's a bit less favourable.
Go ahead.
For integrals, are we allowed to use software to do the integration.
? [01:47:00] Use of computer algebra systems is permitted and encouraged.
Instead of showing my integration working, I can just use a computer algebra system so that I get the results.
That's fine.
In fact, I think it actually says that at the start of the quiz.
I'm sorry.
Just double-checking because that'll cost down my working quite a lot.
All right.
Maybe I can just submit a PDF output from an R Markdown.
That works.
I think also accepts HTML.
Okay.
Good luck.
Thank you very much [inaudible 01:47:38] That concludes today's webinar.

Everybody have a good weekend.