[00:01:00] We're now recording.
The first question was from Helen Barrel, and it was basically about by plots and how to interpret them.
What I'll do is I'll pull up our studio and work through that a little bit for the pizza example.
Then there was a question from Sarah Ray, asking to explain how the variance of y after controlling for x under multivariate normality.
Where does that come from? How does that come from the properties of multivariate normal distribution? Then there was a question from Lily.
I will.
Okay.
Then there was a question on Lily cow.
That was about [00:02:00] Question 1, Part E.
Also check the same Question 1E from Topic 3, testing correlation coefficients.
I think that's all the questions that weren't addressed.
Now.
I'm going to do them in that order because basically the first one can be done using screen sharing, the second one does for screen-sharing done pencil and paper and then the third one is all pencil and paper.
That's why I'm doing them in this order.
Any questions before we start? Actually, sorry, Sarah.
Just to clarify, when you say workings, what do you mean? Do you mean just work through them in the webinar or work through them or something else? I'm not sure how feasible is this just because [00:03:00] actually it takes a long time to put together a lengthy bit of notes.
I do that, but only for some situations where basically there's both there's timing and feasibility.
That particular proof, yes, probably I can't quite think of good keywords to find it.
For example for that question about the variance parameter of the normal distribution, there, you know what to look for.
But again, I think when I go through this on paper, I think it's reasonably straightforward.
For the other one, I think it's pretty straightforward.
I don't know whether anybody actually bother to derive it.
Now, the [00:04:00] first item is, there's our studio.
You should be seeing our studio now.
This is the solutions for the principal component analysis example.
Can I just confirm that everybody can see it reasonably well? So this was just a random file.
Let's start by loading some packages.
There are going to be useful to us [00:05:00] in this analysis.
I'm going to go ahead and actually run through all the code chunks because up until we get to the actual principal component analysis, I can scroll down and then run this one.
So this should load the data set and it has loaded the data set an extra forget about the wine thing that was not supposed to be there.
We ran the principal component analysis, and what it gave us was the standard deviation associated with each component and then the rotation associated with it, and the summary function just gives us a brief summary [00:06:00] that's intended to help us determine how many components we need.
We can also use a manual implementation.
This was just to create a nice picture.
That's odd.
The command use summary and then whatever object used to store principal component analysis result.
So if you call the PC, then that's what it'll be.
Does that make sense? [00:07:00] So for some reason that's actually having a bit of trouble plotting it.

I think what's happening is that it's expecting everything at the same time.
So let's try that now.
Because this is our studio, and it does everything differently.
Yes, this is the plot that results.
We actually make two plots here, one is the cumulative variance explained and one for [00:08:00] the individual variance explained, and one of those cumulative one we use for the cumulative variance method and individual we use for the Kaiser's rule.
Now, I think plot PC does something a bit different, doesn't it? Let's try that.
Yeah, so what this gives you, is the individual variances, which you can use, but that's more useful for the method.
Let's give that a try.
[00:09:00] That's interesting.
I see.
So what this gives you, first of all, it gives you the variances and not their proportions.
But if you want to use one of those criteria, we need to use proportions.
That's small l.
It looks like I just did.
That's what I got on the panel, is not what you expect.
Is that what you expected? Yeah.
Okay.
The reason I prefer this plot is that it puts you on a proportional scale, and as a result, you can use the 90 percent rule by drawing a line at 90 percent, you can use the Kaiser's rule for drawing the line at 1 over the number of variables.
[00:10:00] Although there might be other options, let me see.
It's not being very helpful, is it? This is the method that's actually doing script scope scree plot.
Again, it's good if you can see a sharp drop-off but not so much where you can't.
But the point is that this is mainly for determining the number of principal components and the question was, well, what about biplots? Let's take a look at the biplot of the first two components.
[00:11:00] You can't see that.
What does this actually do? Well, what it does is that it plots the first principal component of each data point against the second, and we have our eigen decomposition to obtain the eigenvectors for the principal components, which we can then multiply each variable by the eigenvector to get the value of the principal component.
Now, we then [00:12:00] plot these values for the first principal plotted against each other but we can also plot the values of the eigenvector essentially against each other.
If I scroll up here, you should be able to see some resemblance, although there's some scaling there so they actually fit on the plot.
But for example, for the first principal component, this is the horizontal axis, we see a high positive value for carb, small positive value for ash and small positive value for sodium, then negative value for protein, then a more negative value for cal and an even bigger [00:13:00] negative value for fat.
What the horizontal axis values of these arrows are, are just these principal components, first column.
Second column gives you the vertical positions, again, scaled so that they actually fit on the plot.
For example, sodium is the biggest negative one, followed by ash.
This is how this plot is constructed.
Make sense so far? I think it's actually one of the [00:14:00] nicer visualisations in multivariate settings.
But why do you say it's clumsy? It's interesting.
Why would you say it's clumsy? Oh, you mean the observation indices? Yeah, that is a downside, I guess you have to zoom in and it mainly works when you have relatively small sample size.

Frankly, to me, the more important part of the biplot is, at least for interpreting the analysis, would actually be the arrows because they actually tell you what each component represents.
Now, [00:15:00] I believe I had some discussion of that down here.
But where we can see is that you have an opposition between carbohydrate content and fat content and which is associated with calorie content and protein content also contributes to calorie content.
My interpretation will be just that fat and protein are more calorie dense.
This is the calorie content and then you have the vertical axis, the second component, which it seems to be the amount of sodium and ash and that again, is probably the amount of dough in the pizza, [00:16:00] although it's not clear.
Maybe this means thicker crust.
I think you do have a slight ash is a little bit tilted, a little bit in the direction of carbohydrates.
I wonder ash it's what happens when pizza burns.
I'm thinking when proteins burn and fat burns, it might bump, I don t think it produces ash.
I think it just produces carbon dioxide and water vapour whereas maybe some of the carbohydrate content burns into ash, I don't know.
Maybe some of the stuff that contains sodium rather burns into ash and maybe to some extent carbohydrates, I don't know.
Anyway, that's how we would interpret this particular plot.
[00:17:00] Does that help, did that makes sense? We can also look at the biplot 1, 2, 3 just to get an idea of that.
Unfortunately, we only have 2-D plots here.
But this one is actually protein versus everything else.
I'm not sure whether the person who asked the question is here but that's okay if not, that's why this is recorded.
But I think that's what I have to say about biplots, I hope that helps.
Now, I'm going to switch up to [00:18:00] the next topic.
Now the next topic was a request by Sarah Ray to basically derive the conditional variance of the estimator so let's do that.
I need to share in the right panel here.
We're back to what is probably our favourite page of this point, which is our single favourite slide, which is the properties of the multivariate normal distribution.
[00:19:00] Just the question is, how do we get to here, basically? If you have the best predictor under multivariate normality, how do we get to this result? I'm going to end that.
That also shows up in terms of when it comes to do multiple correlation where this divided by the original variance of y is 1 minus R squared.
Where does which the document camera now having reminding [00:20:00] you that we'll be using this result.
Well, actually, we'll be using this result and the others are just corollaries.
Sharing the camera now.
Hopefully, good.
[inaudible 00:20:20] your lamp.
The result is true for multivariate normal.
It may be true for other distributions, but here we only short for multivariate normal or at least the only the annotated.
I am going to take [00:21:00] the conditional distribution as a given.
That's a bit proving that it's a bit a scope for the course.
I think some of the materials I linked you can find the proof there.
Also seems a bit unfocused.

That was better.
First of all, what we want to do is we want to predict is we have Y.
Then you have a bunch of Xs, X_1, X_2, etc, up to x and p plus, X_p.
We're going to say that they are distributed normal with a mean.
[00:22:00] Again, we're going to stack these up mu y and then mu x, which is, we're just going to say all these are vectors, x.
What is something we decide.
It's like the disclaimer in other slides this week.
Mathematically, we are talking about correlations that we chose to label these variables y and these Xs, that's how our decision as modellers.
As because perhaps in this case we're interested in predicting y from x rather than this variable, from these other variables, rather than [00:23:00] predicting another set of variables.
In this case though, we're calling it a tau.
Then we assume that distributed normally with mean a mu yx, mu x vectors and variance.
We're going to partition this as sigma yy.
That's just the variance of y.
I think that if, what? Yes, y is univariate our X is multivariate here.
I don't know whether it's clear I use the centre line.
This is a vector, this is a scalar.
I use this under the squiggly underline to make something a vector.
[00:24:00] Because the alternative, in the text I can use boldface.
But it's clunky here and it's also confused with blackboard bold, which looks like this.
You might have seen this symbol for real numbers.
To avert confusion, I use this for to refer to a random vector.
Of course, this for non random vector.
Makes sense? You have sigma y.
Also on that slide we also call it v of y, just as a shorthand.
Then we have the sigma y.
Here we hope we call it Sigma xx.
Then we have [00:25:00] a covariance vector yx, and then we have a same vector transpose.
Just to be clear, this is supposed to be column vector.
It's got a row for each element of x.
Just to make sure we're consistent, I'm going to do that here.
Now further down the page, we're actually using slightly different notation.
We're using Sigma naught to refer to this, and we're using C to refer to this block.
[00:26:00] Now, as we showed using that differentiation exercise also on that slide, the best according to mean squared error estimator for y given x is going to be the expected value of y given x.
Let's write that down.
The expected value of y.
Element in a book or you mean the Johnson [inaudible 00:26:42]? [00:27:00] You know you're right, that's very sorry, I second-guessed myself, but actually I had it right the first time.
You are absolutely right now that I look at it again.
I ended up second-guessing myself and outsmarted myself here.
In fact, yes, so then Sigma naught is a column vector, so it's Sigma naught, which is the notation used in a slide, so yes, thank you and apologies for that.
We want the expected value of y given x, now under multivariate normality, we in fact have that expression from property number 4 and for that we have the mean of the first variable, which is Mu y plus [00:28:00] Sigma 12, so that's Sigma vector transpose xy, I'm sorry, I keep changing this but yes, so

this is Sigma yx, I think this is the last time it's going to happen.
Sigma transpose y, no I'm sorry that was not the last time, I'm sorry, I keep confusing myself here, [00:29:00] so this is xy, not transposed, I swear this is the last time.
This is transpose and Sigma naught this one.
Here we want Sigma, we want the upper right-hand element blocking matrix, which is this one and then times the inverse of xx matrix times x, I'm just going to use lowercase x, specific value of x minus Mu x.
[00:30:00] That's the expected value and we can plug in the other notation that's specialised for the case of univariate result and that was the expected value of y plus Sigma naught transpose vector times c inverse matrix times x minus the expected value of x, so just different notation maybe I should go to that slide just standardise it.
Now, what happens to the variance? Well, we already have an expression for that and that expression is the conditional variance of y given x having some specific value and that's going to be, [00:31:00] we start out with the original variance of y, so this will be Sigma yy squared, then we subtract off the part of the variance explained, so that's going to be, we start out with Sigma 12, which is again Sigma xy transpose.
Then we have the variance covariance matrix of x, again from property for inverse and then you have Sigma xy vector.
Once we translate it to the other notation that gets us the variance of y minus [00:32:00] Sigma naught transpose c inverse Sigma naught, so that's where that expression comes from and that also translates to the coefficient of determination because when we look at the variance of y, given x for some specific value, divided by the variance of y, unconditional, then what we have is this expression we have, so we have this divided by variance of y is just 1 minus the Sigma transpose Sigma naught vector [00:33:00] over variance of y or if you prefer a Sigma yy and that's essentially the proportion of the variance explained by x, so this ratio is the stroke, this is the R squared.
That comes from the expression again property for Sigma, if you look back at that, that's basically Sigma 11 minus Sigma 12, Sigma 22 inverse Sigma 21.
That's just plugging in the specific values for that matrix [00:34:00] property of all the properties, actually we use all of them a little bit.
Does that make sense? Any other questions before we move on to the next question? [00:35:00] Next up, we have the question from [inaudible 00:35:07], specifically Part 1 e on the check your understanding exercises from Topic 3.
I'm actually going to copy this over now to explain.
The idea is that we have our joint tetravariant distribution.
X is normal with mean Mu and there is Sigma.
Actually I'm going to plug them in.
Mu is 1, 2, [00:36:00] 3, 4, and variance is 3, 1, 0, 1, 1, 4, 0, 0, and 0, 0, 1, 4, and last but not least, 1, 0, 4, 20.
This is the scenario we startup with.
This is better? This is just a problem set up because I don't have to flip between the document camera, I'm just going to copy the results over here.
We started out with this trivaried vector X_2, X_3, [00:37:00] X_4 versus X_1 minus 1, 0, 1, 4, 0, 0, 0, 1, 4, 0, 4, 20, that's right? Inverse.
Then vectors X_2, X_3, X_4.
Now, I think I briefly talked about this one last time as well, or maybe the time before that.
I think we talked a bit about how this form should be familiar to you, particularly in light of [00:38:00] one of these expressions.
Because it seems like you have some vector that's maybe taken from here, and then you have this matrix inverted.

It feels like you should have something here.
You've taken x and you subtracted off essentially what would be here, the prediction, or at least this part of the prediction for y.
In fact, one might be able to answer this question in one of several ways.
One way to actually answer it might be more algebraic.
[00:39:00] This is not the solutions provided in the solution, but the solution is basically just calculating it out, performing the inverse and showing that the covariances are all equal to zero.
Now, I think what I will do is I will talk through the first covariance and the rest should be pretty straightforward as well.
You have a covariance of x_2.
This is from the solution page 3, covariance x_2 to x_1, yeah, let's do that one, minus x_2 over 4.
[00:40:00] This second value is obtained just by matrix multiplication.
4 over 4 plus x_3.
This expression is basically is what we get from taking this and plugging everything in and just evaluating it.
Then how do we solve this? Well, for that, we can go to one of the extra slide I inserted in the week where we talked about [00:41:00] the bi-linear property of covariance.
There, the idea is that we can write this as the covariance of x and then some linear combination of a bunch of other x's, we can write it as follows.
Covariance of x_2.
I'm going to be lazy, and I'm going to just use indices here just because it's less time to write.
Sigma 2 1, so covariance between two and one, minus the covariance between Sigma x_2 and x_2, so the variance of x_2 over 4, then minus Sigma 2 4 [00:42:00] over 4, then plus Sigma 2 3.
Now let's plug in the numbers.
The variance between the first and the second variable is one.
The variance of the second variable is 4 divided by 4, that's 1, so 4 over 4.
Minus Sigma 2 4.
Sigma 2 4 is 0.
Plus Sigma 2 3.
Sigma 2 3, that's this element, is zero.
The result is zero.
That's how we would show that.
Then the rest work exactly the same way.
Now, there are some more general results you can show.
[00:43:00] Since we do have some time, I'm going to go ahead and prove it.
The idea is that let's say that we have x_1 and this time it's a vector in the notation of property 4.
We have x_1 minus, and then Sigma 1 2, Sigma 2 2 inverse, and then minus, and then x the second batch.
Next I'm going to be lazy here and I'm just [00:44:00] not going to use parentheses.
The original property use parentheses, but these are vectors, so there's no ambiguity.
This is the first group of x's, and this is the second group of x's.
This is basically the property number 4, but it's a property number 4 without the Mu stuff.
If we just ignore that, that's the property number 4 expectation we also see it here.
We just take this bit and forget about the Mu.
This is the setup, and in fact, you can see how it corresponds to this calculation here.
You have Sigma 1 2, you have Sigma 2 2 inverse, and then you have x_2 and you have x_1 here.
Now we have this expression, and I want to know what is the covariance of this expression [00:45:00] with the original value of X_2? Let's use the same trick as before.

We have, again, I'm going to use Sigma, the covariance between x_1 and x_2.
So Sigma 1,2 minus, now we have something times x_2, and then covariance with x_2.
If you remember the formula, that Sigma 1,2, [00:46:00] Sigma 2,2 inverse, and then we plug in the covariance between x_2 and x_2.
Well, that's Sigma 2,2.
This bit comes from taking the covariance between x_1 and x_2.
This bit comes from taking the covariance between x_2 and x_2 and multiplying it by the coefficient here.
Make sense? Now, you can see what's going to happen.
These are going to cancel out in to identity matrix.
So now you have Sigma 1,2 minus Sigma 1,2, which is 0.
This is a more general result, and this is the other way to prove [00:47:00] this exercise.
That concludes any more general case.
Incidentally, this also proves that in the context of linear regression, the residuals are uncorrelated with the predictor variables.
The reason is because this is our response variable, this is our predictor variables.
I think that's all the questions folks have posted to the thread.
Maybe I can check whether there's anything that's emerged in the thread since this recession started, but I don't think there was.
No, it doesn't look like it.
Well, I guess I will now [00:48:00] switch to consultation mode and take out the questions.
Hi, Pavo.
It's Jay speaking.
Hi, Jay.
Hey.
I just wanted to follow up about the question that I asked in the forums about learning components and the eigenvectors.
I'm a little confused because when you run PR comp or printcomp.
r, you get your vectors.
But those vectors seem to be the same as the vectors that I get when I take the eigenvalues of the covariance matrix, but the coefficient magnitude is the same, but then they are negatives of each other.
When I take the eigenvectors versus when I use PR comp.
I am little bit [inaudible 00:48:56] about why that is the case.
[00:49:00] The answer is that when we do an eigendecomposition, we specify some things about it.
First of all, remember when we have eigen decomposition vector, move to the next page.
Remember, the idea of eigen decomposition that will then give an eigenvector, is that you have some matrix M and the idea is that m times e equals Lambda times e.
That's the definition of an eigenvector or one of them anyway.
Where lambda is now the eigenvalue.
Now the thing is that, notice that if we take [00:50:00] e and multiply it by 2 this expression will still hold, and if you multiply by minus 1 it will still hold.
When we do these decompositions, we impose an additional constraint on e, there are different ways to write it down, but say e transpose z or we should prefer the norm of e equals one.
Make it square norm or not, doesn't really matter because it's just one.
We impose additional constraint, which narrows it down a lot.
You can no longer just scale it arbitrarily.
But you still have the problem that if you replace e with negative e, this relationship [00:51:00] just still holds, and this relationship also holds, right? Yeah.

It's a bit arbitrary doctrine negation.
What that means is that when you perform principal component analysis, your eigen composition can come out arbitrarily one way or the other and it wouldn't make any difference.
The only difference it make is that it will flip one or the other axes of the byplot.
Yeah, that makes sense.
But then for the quiz, we're asked to get the coefficients of your principal components, doesn't that depend on which way your eigenvectors have been flipped? Yes.
There are going to be two correct answers.
Cool.
Perfect.
That's just what I wanted to clarify.
Thank you.
[00:52:00] All right.
Any other questions? In that case, I'm going to switch into consultation mode, or worse about do for a 10 minute break.
I will take a 10 minute break and then we'll switch into consultation mode.
[00:53:00] [00:54:00] [00:55:00] [00:56:00] [00:57:00] [00:58:00] [00:59:00] [01:00:00] [01:01:00] [01:02:00] I'm back.
Several of you just double-check to make sure we're on the same page here.
[01:03:00] That is correct, no worries.
Typo, it's J again, my next question is with the scaling of the matrix or we just, we're just scaling up PCA or principal components, does that mean just normalising them? If that's the case, is that normalising the covariance matrix that we originally [01:04:00] had, basically taking the correlation matrix? It's a second one and specifically, we're scaling the covariance matrix where instead of using data for using the correlation matrix.
Now, that doesn't mean that the principal components will be scaled in the same way.
In fact, they could change in some very complicated ways.
But it does mean that we're converting a covariance matrix to a correlation matrix.
Because you're just taking the eigenvectors for this newly scaled covariance matrix.
That's right.
Perfect.
Makes sense, thank you.
[01:05:00] No worries, you're very welcome.
[01:06:00] The answer to your other question let me just to make sure can you remind me which question that is? You lose track.
A big Sigma hat? I've posted in response some derivation [01:07:00] notes.
I think I posted several of them.
Did you see that post? I think I can let's see.
Looks like I live directly toward that.
There are basically two ways to do it.
One is using some eigenvalue tricks, and the other is using the matrix derivatives, and [01:08:00] so there they are.
Unlucky, surprised how good the Wikipedia's derivation was, how detailed.
[01:09:00] All right, good luck.