Everybody, welcome.

Today we will talk about canonical correlation examples,

and the software demonstration there will be using the standard set of packages.

But the particular new ones will be talking about today is the CCA package,

which as you can see, has a lot of dependencies,

let's not worry about that,

and the CCP package.

Now, the data we'll be dealing with here is the fitness club data.

The idea is pretty straightforward.

They take three physiological measurements of

a person for each of the 20 middle aged men in a fitness club.

Those three physiological measurements are the weight,

I believe in pounds, waist in inches,

and pulse in beats per minute.

Then there were three exercise measurements, three exercises variables,

the number of chin pull-ups,

the number of sit-ups,

and number of jumps.

Again, I believe it's per some period or maybe before they could no longer do anymore.

The way we'll use canonical correlation here is to

determine if these physiological variables

are some way related to the exercise variables.

We'll start by loading the data.

You've seen this construction before.

We have essentially a string of texts containing a table and

then it gets passed onto a text connection to make a fake file out of it.

Then we read a table out of it and we assign the result.

This is the forward assignment operation to fitness,

and this is what the data set looks like.

Now, here are the variables,

here are the summaries,

as you can see they have different means, different variances.

But of course, what we're most interested in is their correlations and in particular,

we can see there are some associate correlations between them,

both within each of the group of three and between them.

Also there are some outliers in the data.

As always, I would encourage you to work through this example on your own,

that is in R, and also maybe see what happens when you remove the outliers.

Let's actually fit the canonical correlation.

The function that we will be using here is the function CC found in the CCA package.

It's pretty straightforward.

We're going to create one matrix of data frame with the first three variables,

the physiological measurement,

and another one with the secondary variables,

a data frame with the exercise measurements.

Then we're going to pass them both to the CC function,

assign them to a variable tool called fitness.cc and we'll go print the result.

This is what it looks like.

You can see some immediate elements of the output elements.

The first thing we see is the actual canonical correlations,

the first one, the second one,

and the third one.

Then we see the names for the first set of variables and the second,

for convenience, the indexings for the observations.

Then we have the coefficients for the canonical variables.

These are of course the a and the b that we derived earlier.

How do each of these variables map onto their respective canonical variates?

Then we have their scores,

that is the value from that mapping and then we have some useful summaries,

the correlation between X variables and their scores.

That is how each variable relates to its score.

Correlation between Y and the scores of x,

the correlation between X and the scores of y and

the correlation of Y and the scores of y.

Now, why would we want to do that?

What is the difference between, say,

correlation between X and x scores and these coefficients?

Well, the thing to remember is that these variables,

weight, waist, and pulse are correlated.

We can see that up here, in particular,

we see that weight is of course highly correlated with waist.

There may be some inverse correlation with pulse.

We don't know the sample size is pretty small.

The idea then is that since these are positively correlated, first,

when you actually take a linear combination of these,

the correlation of the result with these variables might not be the same.

Now, then again you can use this to understand how these variables relate to each other.

This is also summarised here.

One thing we can do is then using the facilities in that packages

to plot some meaningful representations of these canonical correlations.

The function that does it is a bit hard to remember,

but you don't have to memorise it.

It's plt.cc, we'll also use va.label=TRUE,

which we'll label which variates have which canonical weights.

What we see here is that on the left-hand side plot is the correlation

between X and the xscores for the first two canonical vectors for each of the variables.

Then the correlation between Y and xscores,

not Y and yscores but Y and xscores.

What does this tell us? Let's focus on the first canonical vector for now,

that's the horizontal axis.

The values with the biggest magnitude there are waist and weight.

The first canonical vector,

essentially the smaller it is,

the higher are the person's waist and weight,

hence the negative correlation here.

As they increase and negatively correlate in particular with chins and sit-ups,

we'll have a strong positive correlation with the dimension 1.

We can see again, positive here,

negative here, and something relatively small in magnitude here.

We would say that the first canonical correlation represents that people

who are overall heavier tend to have fewer chin pull-ups and sit-ups.

It makes sense. The second one,

as always, it contains whatever the relationship is over and above that,

so it's going to be a bit hard to interpret.

Here, we do see that there is a positive value for weight in

particular and a negative value for jumps.

Maybe that's a relationship perhaps after accounting for weight,

waist does not really associate with jumps, whereas weight is.

A right-hand plot refers to the individual variants,

and I think we can see the outliers here as well.

Under some circumstances,

particularly if you want to compare the coefficient's magnitudes,

it can be helpful to standardise the variables first.

This is pretty straightforward,

all we have to do is to use the scale function.

That was introduced in the video demonstration on matrices earlier in the course,

but here I generate x-scaled and y-scaled then we fit.

One thing to keep in mind is that,

for hypothesis tests you should use unstandardised data only.

We have the canonical correlations here

and the coefficients and everything is the same.

The idea is that because we've standardised the variables,

that is, we've centred them at zero,

which doesn't affect correlations and set their variance to 1,

these can in some sense be interpreted in terms of

Z scores or standard deviations from the mean,

rather than the raw scales.

The other thing to notice here is that

the actual canonical correlations are exactly the same.

Moving on, hypothesis testing.

Hypothesis testing, as we saw in the slides,

actually depends on very little.

It only requires correlation, sample size,

and how many variables are in each group,

and of course, information about the null hypothesis.

The function that does that again,

in the CCP package, is the p.asym.

It requires again, the sample size,

the number of variables in the first set and number of variables in the

second and of course the canonical correlations themselves.

You see the canonical correlation fit from the start and we pass them here,

and then we pass the n,

p, and q, and we specify.

For the moment, let's use the Wilk's test,

that is the Wilk's Lambda statistic or essentially,

the likelihood ratio test.

Here are the p-values we get.

We actually get the idea that now these hypotheses are basically saying that,

and again, this is called a discussion.

In the slides, this represents the null hypothesis

that not all of the canonical correlations are zero,

that is, the population.

If we were to take the values of a and b,

the coefficients that maximise the correlation between the results of

multiplying the coefficient a by

the variables in the first set and coefficient p by the variables in the second set,

we take the correlation between that,

make it as big as we can,

the population correlation is still zero,

not sample, but in population,

that's null hypothesis represented here.

The other interpretations, of course,

that there is no true linear relationship in the population,

between any of these two sets of variables or any of their linear combinations.

The second hypothesis only refers to setting

the second and third canonical correlation to zero.

That hypothesis is that after accounting for the first canonical correlation,

correlation due to this first pair of linear combinations,

there is no left over linear relationship between the variables,

that is, between the two sets of variables.

Same thing with three to three,

which is that, there's after counting for the first two.

You can use the Hotelling's criterion as well.

It seems to say that the first canonical correlation is significant, the others are not.

Pillai's trace or significant at the conventional level of 0.05,

because of course, p-values have their issues.

We can also get the Pillai's trace where the Roy is Greatest Root,

which is only testing

the first canonical correlation and it is the least conservative test.

Part of the problem here is that, remember those outliers,

we do have deviations from normality and we have a relatively small sample size.

That also means that tests don't necessarily agree.

One way we can get around the fact that we don't have normality,

we have outliers to show permutation tests.

The idea there is that,

we're going to take essentially one of the sets of variable,

say X, and hold it fixed,

and we're going to take the second set,

say Y, and scramble it.

We're going to randomly permute its rows.

Under the null hypothesis,

when there is no canonical correlation,

it wouldn't make any difference because we know before scrambling, after scrambling.

When we scramble them like that,

we get the sampling distribution under the null hypothesis,

which lets us get these p-values.

As you can see, the p-values tend to agree a lot more,

although one limitation is that it's a bit harder to test for correlations beyond that.

But anyway, the bottom line is that these tests are

valid even if you don't have normality,

though do keep in mind that we are testing linear correlation in particular here.

The last example when only correlations are available.

Here we have four variables.

The first pair is the skull length and breadth,

and the second per variables,

there's femur and tibia length and there were 276 chickens.

We want to see how this calculation can be done without the original data,

but only using the correlation matrix.

We'll load the correlation matrix here, the usual way.

We'll also give it column names here,

so it'll look nicer and we assign the sample size here.

This is basically the implementation of the materials in the slides.

It's implementation formulas in the slides,

I'm not going to go into details.

You've all seen eigen decomposition and we have the matpow function.

We could have also used the pow 1 package.

The matpow function is just the eigenvalue-based matrix power.

The idea then is that we have our eigen decomposition

and from that we get the canonical correlations by taking the square root of

the eigenvalues and from the rest of it we get a and b.

Again, we use the CCP packages,

p.asym, which by the way,

is where this function p.perm comes from.

We can then do the tests using the R package,

p.asym, and here they are.

The first canonical correlation is highly significant.

We can also interpret the first canonical correlation here.

They are the correlations,

these are the coefficients,

this is the coefficient for

the head variable measurements and these are the coefficients for the leg measurements.

This concludes this demonstration.