Everybody, welcome.

I will now go through the cluster analysis demonstration.

As always, I would encourage you to work along in instances of your own.

The special packages we'll need for today

is the candisc package for canonical discriminants,

we'll talk a bit about that later.

The package cluster,

which has a lot of different clustering methods,

and the package mclust which implements model-based clustering.

We will as always be using the Iris example.

Suppose for a moment that we

have mixed up our lab samples and no longer know which specimen comes from which species.

The question is, can we recover the species?

This gives us a baseline to see how good a clustering might be.

Now, we'll also define,

in other words have an activity Iris not which will drop,

this is the species column.

Now, the plotting function down here is

going to be useful for visualising the clustering.

We'll plot the Iris data in

pairwise and then plot

the two clustering by the character hour by the plotting symbol,

and use the colour to code the cluster assignment.

We will also use something called canonical discrimination.

The idea there is that we're going to fit essentially a linear model

for clustering variables on

the left-hand side and on the right-hand side we'll have the clustering,

and then will run the canonical discriminant method on the fit.

Again, we'll talk about that later when I show you the plots.

Then we'll plot them and we'll use a certain encoding

for the colour-coding the clustering.

One thing we can do is we can use hierarchical clustering,

does a built-in function stats package called hclust.

It uses the Ward's method with squared Euclidean distance,

or rather it's something we can use.

You can use others as well.

Here we have the fit,

I noticed that it doesn't take the original dataset,

but rather a matrix of distances which are obtained using the dist function.

We have to specify the method and then we can plot it,

and this is a dendrogram and the height refers

essentially to where you set your threshold for declaring,

we have enough clusters.

The idea then is also

how much squared variation do we gain or lose when we split or join a cluster.

Let's cut it at three clusters,

cutree is the name functioning that does that.

Here's our clustering,

here's the confusion matrix.

Here, can we recover the species?

Well, turns out quite well actually.

We get particularly, I think,

because we have Petal.Length and Petal.Width here, which are very good in that respect.

We have all also the canonical discriminant here.

What does it tell us? Well,

it tells us for one thing that pretty much all variables are

redundant to each other from parts of the clustering relative with respect to each other.

What we see is that in particular Petal.Length and Petal.Width are really important,

but also they are a little bit redundant.

Sepal.Length contains maybe additional information.

Although top there how much and same thing with Sepal.Width because one thing you

can see here is that this is 97 per cent,

this is two per cent.

Really the additional value of Sepal.Length and Sepal.Width is minimal.

We can also use cluster package which is more flexible.

The function is called the dose hierarchical,

clustering is called agnes or agnes.

Again using Ward method, you can look up,

it's a lot more flexible than what's built into stats.

Here is what the clustering looks like,

but it's really the same thing.

One thing that it does give us,

this package cluster, is the silhouettes.

We can ask which cluster number gives us the best average silhouette width, and

we see that the actual silhouette width seems to actually decline throughout,

although it does seem to actually prefer to 2-3,

because the highest silhouette is actually have got two clusters.

This is because perhaps two as together versicolor and virginica alas.

On the other hand, if we do know how many clusters there are,

we do get a very good recovery.

We can also use non-hierarchical clustering.

We talked about k-means in the slides,

that there is a function built into the stats package.

Here's what it produces,

it's a result that's very, very similar.

Again, it actually favours two clusters rather than below,

it says maybe there's an argument for five, I guess.

k-medioids, the function that does

that is in the cluster package units called pam partitioning around medioids.

Yes, a lot of the functions in the cluster package are given

these slightly outmoded female names.

But here it is,

the clustering, and notice that this plot is essentially the same,

but it's been flipped just because these are arbitrary,

like which cluster corresponds to which category?

Again, silhouette plots, same.

Last but not least, model-based clustering.

One thing you can do it for model-based clustering is to just give it

the dataset to cluster and it'll try to figure out the best model.

If you recall the slides,

it'll tell you that the best model was varying sizes,

equal orientations, and varying rotations.

Now, it does select the two clusters and it does blur virginica with versicolor.

We can't force it to use three clusters,

and it does suggest the same model.

But now, it

does identify the two cluster pretty well and you

can visualise a whole bunch of aspects of this clustering.

The first canonical discriminant is even higher than before,

which is probably a good thing.

The classification actually produces a much,

much, much lower error rate can others.

In fact, only five points are misclassified.

The reason for that is that this model is a bit more flexible than say,

k-means or others that took or rely on Euclidean distance because this type of model can

actually have different dimensions counted differently,

we'll not count at all.

This concludes the demonstration on clustering.

As always I recommend working through

it yourself and looking up to help for each of these functions.