Everybody, welcome.

I will now walk through a tutorial on Basic Multivariate Summaries and Visualisation.

We will use the classic Iris dataset.

This dataset, perhaps classical point of triteness,

comprises 150 measurements of Irises, that's a flower.

These Irises come in three species,

Setosa, Versicolor, and Virginica.

For each of those flowers,

we get a measurement of sepal length,

sepal width, petal length, and petal width.

These are all parts of a flower.

Let's take a look at the basic data summaries.

We will use both the built-in summary functions and dplyr tidyverse tools,

which you have probably seen in other classes.

We start with the output of this data frame, Iris.

This is what it looks like.

We have four measurements and one categorical factor.

Now, first,

we run the summary of Iris and R has a pretty good built-in summary function.

You get the mean minimum,

first quartile, median, mean,

third quartile, maximum and you get the frequencies for the categorical variable.

There are 50 flowers of each species measured.

Now, that's not very multivariate test.

But one thing we can do is to take the means of each column,

that is the mean of each variable and the function that does that is colmeans.

Now note that here I have this minus 5 bit.

That basically means ignore

the last column because that last column is a categorical variable.

You can't have a mean of a categorical variable.

Another useful function that you may or may not have seen before is apply.

The idea there is we take this data frame,

and then we perform some function on a particular margin.

In this case, we say two here,

so then that says for each column,

evaluate the standard deviation.

Here the column standard deviations.

We can also get things like quantiles.

The apply function here, the argument again,

the dataset, the margin column,

a function quantile,

and then in addition to the actual data,

actual content of the column will pass 0.95 to get the 95th percentile.

In term of multivariate summaries,

first thing we can do is sample variance-covariance matrix.

That's pretty straightforward.

That's just a cov function.

It'll take the little computer covariance of all the columns.

Correlation matrix as well.

There is a function called cov 2 cor that's two and a different number.

That'll take a covariance matrix and convert it to a correlation matrix.

We could also apply this to subgroups.

One way to do that is to use the pipe operator from the package magrittr well,

I'll explain what it does in a moment.

We can also use the map function for purrr package,

which does something very similar to another R built and R function called L plane.

But anyway, here are the two libraries involved.

We start out with an Iris dataset.

Then we pass it onto the split function as its first argument,

and with the second argument is Iris dollar sign species,

that means we split it according to its species column.

Split is a function that's built into R that does that for data frames.

Then the next we pass it to the map function.

The map function takes each element of this list.

Now, this is now a list with an element for each species.

This basically says apply whatever is here to the element of this list.

Now the dot is where we substitute the data frame containing

just the flowers of that species,

and then we remove the fifth column.

Finally, we again map and for each element,

for each species, we compute the covariance matrix.

What I would suggest when going through this is to run just this code,

that is iris and then split and see what it produces.

Then run the first three lines to see what results.

Then finally run the rest of it and confirm that what you

get is this variance-covariance matrix for each species.

The most common way to visualise multivariate data is a pairwise plot.

The idea there is pretty straightforward.

We just take each possible pair of variables and you plot them against each other.

Now, the thing is because different types of variables acquire different types of plots.

It can be inconvenient because for example,

in this case, one of our variables is categorical.

Let's use the ggpairs function in the GGally package.

Here, I'm doing a bit of extra.

If I just pass ggpairs of iris.

I will actually do something very sensible with plot it,

pretty much what you'd expect.

But we can actually do a little better,

and what I'm doing here is I'm saying aes that's stands for aesthetic.

I'm saying let's colour

the plots by species and also will make the points and whatever else,

a little bit transparent.

Alpha 0.7 means they are about 70 percent opaque.

Alpha equals 1 means a point is plotted opaquely,

and Alpha equals 0 means that they're plotted transparently.

What we see then is this on the diagonal.

We have the density plots for each of the variables.

Note that because they're colour-coded and semi-transparent,

we can actually see where these distributions overlap.

Then in the lower triangle here,

we see the left-hand rather,

we see the scatter plots of all the groups and we can inspect which variables,

for example, separate these groups the best or which combinations of variables.

Again, these are pair-wise plots.

Then in the upper right-hand side here we

have correlations and we have both the overall correlation,

but also correlation within each species.

By the way, an interesting observation here is that

the correlations within each species are actually positive.

But the correlation overall is negative.

When we combine all these datasets,

the correlation is negative.

This is actually an example of a logical fallacy.

We won't be getting into it in this course.

It's a bit outside the scope,

but I would suggest looking it up it actually shows up in a lot of different places.

So far, I've discussed the quantitative plots only.

Let's look at the categorical ones.

Now if here we have species on the vertical axis

and the measurement on the horizontal axis here

we just get these histograms broken down by species, convenient.

Here you have one bar-plot giving the frequency of each species.

In this case, there're 50 of each so there isn't really a difference.

Lastly, here we see plots of each of the measurements,

again broken down by the species box plots,

which give us some idea of the distribution and the overlap as well.

In some sense, these are telling us similar things to these and similar things to these,

and the reason is because we are colour-coding them here and also using them here.

If we didn't colour-code them,

then we wouldn't get as much detail here,

but we still would get the species breakdowns here.

Again, I would suggest playing around with this statement,

perhaps see what happens when you get rid of the aes.

Perhaps see what happens when you have just the parenthesis.

Well, this concludes this demonstration. Good luck.