Everybody, welcome.

Today I'm going to go through the demonstration for the multivariate normal distribution.

What I would recommend when you follow along,

is to start up your own studio session,

there will be one in the window and to just copy-paste to

the bits of code as I talk through them and see what they do.

Now, the output is given to you here so you don't have to do it,

but I think it'll help.

I also recommend just playing around with it.

Now, we'll be using a number of our packages which would be

loading from the start. Here they are.

The first one is here,

this is a convenient library for keeping track of your files that you

put all your data files in one directory and then just reference to here,

then subdirectory, then the data file.

MVT norm is a package for managing multivariate normal distributions.

GG ally is a toolkit

for making a variety of plots using the GG plot 2 interface.

MVN is a package for multivariate normal testing and diagnosis.

Dplyr is a tidy verse data management package and readr

is a tidy verse package for loading and saving data files.

Let's play around with multivariate normal random variables.

The first thing we can do is we can generate them.

Now for that, we have package MVT norm here

and we have functions DVMT norm for density,

RMVT norm for random values, and so on.

This is a pretty standard notation used,

we'll it demonstrate it shortly.

First, let's construct a vector mu for the mean,

it's just going to be vector 2, 3,

4, here it is.

Now the way we're going to have to construct Sigma is we're going to create

a diagonal matrix whose diagonal elements are 1, 2.

3, and then add 1 to every element of that matrix.

Now, that gives us a matrix that looks like this.

What happens when we generate the values?

Well, it's simply RMV norm,

the number of values to generate the mean and the variance matrix.

Then we end up with a 1,000 by 3 matrix,

with samples in rows and variables in columns.

Let's plot these variables.

A useful function for that is the function ggpairs in the package GG ally,

which we loaded up here.

First, we need to convert this matrix to

a DataFrame so that we have all the variables in columns,

and it is going to give us default names for those variables or columns V1, V2, and V3.

It gives us this plot.

We could also use a different function called pairs,

but it won't get quite as rich in output.

Here we get density plots for each of

the variables from the diagonal and then below the diagonal and the lower triangles,

we get the scatter plots for each pair of variables.

We can see there's some correlation here and in fact,

we get to see the correlation in each of the quadrants here.

Now, these correlation values,

you could actually derive them from the form of this matrix.

Try it now. I'll give you a hint.

There's a function called COV 2 COR,

that's COV the number 2 and then COR.

That will simplify that process for you.

Now, one lesson that we talked about in the lecture was that

marginally normal distributions are not necessarily jointly normal.

That is, they have two or more variables,

each of which has normal in and of itself,

that does not mean that they're jointly multivariate normal.

This is a pretty standard example that you will be able to see,

in fact, I think it's from a Wikipedia article on Normal Distribution.

Let's take two variables.

We'll generate as follows.

First, we will generate a variable Y,

which is just a standard normal random variable univariate.

Then we'll generate a variable W.

Now, W will be generated from

a binomial distribution with one trial and probability success of 1/2,

so a coin toss basically.

The idea is that it has a value 0 with

probability 1/2 and value 1 with probability 1/2.

Now, notice that that means that if we take

this variable and multiply it by 2 and subtract 1,

we'll get minus 1 and plus 1 with equal probability.

Then let's generate our two variables.

Z_1 is going to be equal to Y and then multiplied by 2W minus 1.

It'll be wide but with the sign flipped randomly.

Z_2 is just going to be a copy of one.

I'm going to run the R code now that would generate it.

First, we generate the Y and then we generate the W, 1,000 of each.

Then we use Cbind to create a two-column matrix.

The first column is 2 times W minus 1 times Y,

and the second column is 1,

so Z_1 and Z_2.

Here's what they look like.

Now, V1 is normal,

you can see that from the histogram it's pretty close to normal,

or from the density plot, sorry.

V2 or Z_2,

that label is Y by default here, is also normal.

But together they are not normal.

In fact, they formed this weird cross shape.

We can see what jointly multivariate normal looks like here.

It's going to be like a football shape,

but not a cross.

Now, let's see what happens when you actually run diagnostics on this.

We'll use MVN package.

It has a mvn function that performs all sorts of multi-normality diagnostics.

Actually, the mvn function has a variety of

different diagnostics and they

reflect the package author's preference about which tests they think work better.
In particular, they favour the Henze-Zirkler test for
the joint multivariate normality and Anderson-Darling test for univariate normality.
We haven't discussed them in the course materials,
but they are relatively more recent tests,
in fact, the package used to have the others as a default.
Now, regarding those,
here's how we could ask it to compute
the Mardia diagnostics and the Shapiro-Wilk diagnostics to have you more familiar.
What we see and I think this is pretty helpful,
first of all, for univariate tests,
we actually see basically the same conclusion that the tests
of the distributions are probably univariate normal.
What Mardia test lets us see,
even though it might not be as robust as Henze-Zirkler is that,
well, according to the skewness measure,
it's consistent with normality,
and we can see why,
because this is a symmetric distribution.
Whereas according to kurtosis,
that is how heavy the tails are,
it is not normal.
What we see is that individual dimensions all show up as normal,
but jointly they do not.
One property of a normal distribution is any projection of
a multivariate normal distribution is going to be normal.
Let's do that now. But this is one projection,

where we use the vector 1 and minus 1 and then we use matrix multiplication.

What that means is that we're going to take the matrix and essentially take a sum of

1 times z1 plus minus 1 times z2,

which gives us putting this variable and plot it to density.

We see that we get something that is not quite normal.

You don't have that nice bell-shape, you see you have a big spike in the middle.

Whereas if we have our original x,

we generated it for the from multivariate normal.

Here we picked the same coefficients and we do get something bell-shaped.

Again, it's a bit ugly,

but it's close to normal.

If we used N,

we would have gotten something even more normal-like.

Or similarly, we could use a random projection here,

just pick three random numbers as coefficients and we get something normal-ish.

One example I'm going to go through now is from Johnson and Wichern,

which is a suggested textbook for this course and

it consists of measurements of radiation made from

42 randomly chosen microwave ovens with doors open and closed.

Below the dataset, this is where we use read_csv from the read R package.

This is an application of the here function.

I've put the dataset ovens.csv in the dataset's directory.

That itself is inside the course directory.

The here function can find the course directory then tax on

datasets then ovens.csv and loads the dataset.

We're told there are 42 rows and 2 columns in it,

and both of them are doubled, that is, they're numeric.

This here is what they look like.

Now, these are not normal at all.

We can see that they are correlated,

which makes sense since we're talking about the same ovens.

But clearly there's some kind of right skewness here. What do we see?

Well, the default tests,

they all show that things are definitely not normal,

same thing with the tests we've discussed,

neither skewness nor kurtosis are consistent with the normal distribution.

Now, let's try transforming the data.

Now, when we transform the data,

we need to be careful because

transformations can change the interpretation of our results.

Nonetheless, let's try to do that here.

We now start transforming the data.

One thing to keep in mind always is that when you transform the data,

you are actually changing the model,

you're fitting in the interpretation of the results and all that,

so we do want to be careful in interpreting our conclusions but for now,

let's go ahead and look at what the diagnostic say.

Here we transform by taking the fourth root again,

has more details on that.

Now we get something that's a lot more sensible looking.

Maybe the open variable,

that is measurements when the microwave ovens are open,

they might have slight by modality or right-skewness.

Let's now try the test.

What we see is this.

Now, the Henze-Zirkler test says that

there's some evidence of non-normality not as strong as there was before.

In particular things that may be the marginal distribution of open is,

in fact, maybe not normal.

A similar Mardia tests is not as powerful,

it concludes that in fact,

both skewness and kurtosis are consistent with normal.

Although it also says that well,

but marginally may be Shapiro-Wilk is not particularly for open ovens.

This concludes this demonstration.

Go ahead and play around with this and try the challenge exercises next.