- In this lesson, we'll be focused on the value of Numpy,

the primary package for scientific computing.

By the end of this video, you should be able

to recognize the value of Numpy for data science.

Numpy offers a number of key features

for scientific computing.

The key feature of Numpy is support

for multi-dimensional arrays.

This is useful for representing vectors and matrices.

When doing data science, we'll be working

with matrices nearly all the time.

Numpy also gives us a number of operations

which we can perform on matrices.

This includes obvious things from linear algebra

like adding, multiplying,

subtracting of matrices and vectors,

but also includes optimized statistical operations

fast Fourier transforms among others.

One of the trickier things to manage

when working with matrices and vectors

is making sure their dimensions align properly.

Numpy actually takes a lot of that work out of the process

by supporting something called broadcasting

which makes the process easier

and the code easier to read.

Lastly, Numpy is often fast enough for production code,

so you don't need to optimize further,

but if you need to, it gives you the ability

to work with optimized compile code from libraries

from Fortran, C, and C++.

This functionality I just mentioned is really useful

but there are really three reasons data science

are working with Python using Numpy all the time.

The first is that it's fast.

In fact, using Numpy's arrays can be 10 times faster

than Python's lists.

To enable the speed, Numpy arrays are fixed in size

unlike lists which can change in size.

In elements and Numpy arrays must all be the same type.

All ints or floats where as lists can hold any type.

By restricting ndarrays in this way,

it makes ndarrays both much more space efficient

than lists, but also opens up a range of memory

and computational optimizations.

The second reason we use Numpy

is because of its functionality.

Those operations I mentioned are really useful.

Want to get the average of a vector I the matrix?

You got it.

Want to multiply matrices?

You got it.

Want to select a subset of the matrix

based on indexes or values?

You got it.

Even when we move into Pandas, you'll be finding yourself

using many of these functions.

Lastly, we love Numpy because many of our favorite packages

in Python rely on Numpy.

In fact, Pandas which we'll be doing soon

are built on top of Numpy.

I should note that while Pandas

provides higher level functionality than Numpy,

you'll still be using Numpy as functionality at times.