Hi! Im shalmon Anandas. A masters student at Khalsa college. If I had to describe myself It would be “The tech guy”. My expertise, you could say is in anything related to computers. I love coding, I like tinkering with hardware. Writing drivers for awkward, rare hardware that I find that doesn’t have support for recent features. I’ll sum it up as “If there is a computer involved, I’ll most likely enjoy it”

So I’ll be covering statistics using python. So starting with a builtin library in python called “statistics”. This is the simplest way to perform statistics in python. I’ll give you an example of how it works, first we import median from the statistics library. Then we import isnan from math and filterfalse from itertools. The use of all of these will become clear as we go.

We create a list of data with NaN values (NaN) values are values that are of non integer and non string type . Now we will try to perform operations on this list. When we try to sort this it presents an unexpected behaviour, that is the list isn’t sorted. When we try to get the median of this dataset we are met with another unexpeted behaviour, that is it doesn’t really give up the proper median. This is because of the NaN values in our list. First we figure out how many of our values are NaN. In our case there are 2 NaN values. to fix this we will create a new list called clean. In this we will use filterfalse to remove values and use isnan as a denote to which value is to be removed and data as our list from where the values will be removed. This will give up a clean list with no NaN values and the previous operations that we tried to do will be executed properly.

There are other libraries for statistics and numerical data manipulation / scientific calculation in python. One of those is numpy. We will look at numpy in the scope of statistics. First we see what can't our inbuilt statistics library do and then see how numpy helps us. So to find median, mean from a dataset that isn't just 1D, statistics library falls short. We will start by creating a data set that has 3 lists inside a list. When we try to find median from this dataset, we get the middle list as our median rather than an actual value. Finding the mean just straight up doesn't work when using an inbuilt library. So now moving on to numpy, we will first create a numpy array which as 3 axes. Each list inside a list is known as an axes in numpy. When we try to find the median, it gives up an actual middle value out of all the whole dataset, and same with finding the mean.

This it the basic, surface level advantage of using third party libraries vs using inbuilt libraries, but, it isn't always that plain and simple. We have to know our dataset, what our requirements are and the scope of our analysis to then select a library for our code. If it's a 1D dataset and we have to do basic simple calculations there is no use for numpy, unless it is a very very huge dataset where in that case the speed of numpy execution will be to our advantage. SO, my point here is that there is no 1 library that does all, its never cut and dry and we need to have indepth understanding of what each library does to use then in a real world scenario