Radio Astronomy

Prof. Abhirup Datta

Department of Astronomy, Astrophysics, and Space Engineering

Indian Institute of Technology Indore

Lec-25

Few Concepts with the help of python as a computational tool

Hello once again to all of you. I am Harsha Avinash Tanti and I am currently a research scholar at Department of Astronomy, astrophysics and space engineering at IIT, Indore and I am your current course TA. So let's begin today's lecture. So today's lecture is about doing some computational examples on python to understand few concepts more clearly. So as for today's overview what we will go through will be what is 1D FFT, one dimensional FFT which you have learnt in week 4 specifically lecture 2 and lecture 3 we will be covering. So what is 1D FFT, what is two dimensional FFT? So basically you might have, sir might have introduced some aperture distribution as well as few examples of that but if you say you have an aperture or an area so that is in two dimension.

And sometimes in three dimension also there is z also but mostly we can assume it is a flat screen and take it as two dimension thing. So in that case and also we studied that in I think lecture week 8 that, sorry week 7 that aperture and the signal obtained is in Fourier transform relationship as well as the visibility and the intensity also. So here we will see how the, what is 2D transformation, Fourier transformation and how we can use, how we can visualize it. Now Dana we will also see what is DFT and FFT.

We will discuss it about a bit later why DFT and FFT. And after that we will learn about something called correlation as you might have also covered in this or the previous week that if we want to go for interferometer technique we have to correlate two signals. So how to correlate that is also we will cover and the additional topic is sampling which is very important and how it affects your data and basically the thing we will discuss here is aliasing. So let us move to the review, a short review and introduction about it. So what we will have here is, so we will cover what week 7 lecture 4.

In week 7 lecture 4 we covered Python, how typical crash process Python and how we can utilize it as a computational tool. That was a bit hasty but we will go through one by one each and every example and I will cover in a step by step way manner what we are doing. So don't need to worry about it. Now also further more also we will visualize what we are computing means like by means of plotting and all those things. Then let's quickly cover the basics means like how you can utilize Python online which is also called Python notebook that is ipynb notebook.

So what you have to do, you have to go to this link here. So this link you have to go where after when you go to this link, this Google Colab that will redirect you to your Google accounts login page. So there you have to put in your Google ID and password and you will be ready to go. Now let's before going to the Python scripting or visualizing our course and everything what we will do, we will quickly revise few concepts. So if you recall week 4 lecture 2 what we covered there how the Fourier series converts into Fourier transform and the general equation is given like this.

But one thing I will say and the inverse of this is this. So these are Fourier pairs. Now here the thing is this is continuous time. Now we will discuss why I said continuous in time. And then what we learned in week 7.

What aperture distribution is Fourier pair with whatever the intensity distribution you are getting. So intensity distribution you are getting these two are Fourier pair. But what you might have seen there we had aperture like this, we had two point apertures like this or we had aperture like this, something like this. But in reality we have a two dimensional aperture means either it will be a circular. Let it be a parabolic dish but however if you see from a top view it will appear as a circular.

So with that the concept of 2D Fourier transform comes. So 2D Fourier transform is there already similar to 1D Fourier transform. But here we will use 2D Fourier transform to see how the aperture distribution looks like in two dimension. In one dimension we already know and we will already cover how the Fourier transforms looks like but it is very interesting to know how the Fourier transform looks in two dimensional.

Okay. So now let us just skip back a bit. Here also you can see this is CT, CT naught or continuous, continuous FT, CFT continuous Fourier transform. And this is also CFT continuous Fourier transform. Now what happens in continuous signals for between two points say I have a signal which is going from x0 to x1 or t0 to t1 or in any space. If it is in continuous Fourier transform between two signal, two values there are infinite number of values which is not, which cannot be stored in a computer or a computational device or a digital device.

So for that we need to sample the signal which is coming, the analog signal. Now if we sample the analog signal and if we Fourier transform it what we will get, we will get something called DTFT, discrete time Fourier transform. But what happens in this? In this you will get a continuous Fourier spectrum. A continuous Fourier spectrum. Again which is not, which you cannot store in a PC or you can't compute and show the result.

So that's why what happens. We do sampling both in Fourier and time domain or the other space domain. So this is how a DFT equation looks like. Now similar to this 2D will be having one more term over here e to the power minus j k omega instead of n that will be m and small m and here will be double sigma. So this way 2 dimensional DFT will be there.

Now a term which you will be frequently hearing is FFT, something called Fast Fourier Transform. Now if you see this equation, okay, if you remember clearly week 4 lecture 2, what you have, you have a DFT matrix then there is a one another matrix which is exponential matrix or this term matrix multiplied along with the signal matrix. Okay. Now if you remember correctly, this is a matrix multiplication which means this multiplied by this. So this will go into, if you program it, if you try to program it, just take it as a word, it will go into loop.

Means there will be multiple loops running to calculate this. To reduce that an algorithm is used to reduce the time consumed for calculating DFT which is called Fast Fourier Transform. So Fast Fourier Transform is an algorithm which computes DFT or inverse DFT. So and there are multiple algorithms in this but the collectively it is called FFT which is nothing but an algorithm which can perform DFT. So here we will be using Numpy, Numpy library in which we will use this particular function and also we will define our own DFT function by which we can calculate these things.

So now let us take a look at correlation. What is correlation? There are two types of correlation, autocorrelation and cross correlation. So autocorrelation gives the how much a signal is correlated to itself. Means like it will kind of give you signal power. Okay if you remember correctly it will be as in Fourier domain it will be X omega, okay X conjugate F omega.

It will be like this, okay so which is a signal power and similarly here you will have a cross power between two separate signals what is the power distribution or how it is correlated to each other. Correlated means how it is similar to each other. Now there is another thing called Wiener-Kingshin theorem which suggests that if you want to save the computation time what you can do you can just use the take first signal do a Fourier transform then take another signal do its Fourier transform and conjugate it and multiply with each other you will get the cross correlation. Now okay so we will stop here and we will revisit the slides again for sampling concept. So let us see how we can visualize these things using Python.

So if you login you will get this kind of notebook here so you have to just connect it so it will allot you, Google will allot you RAM and memory to work with. So what we will be using we will be using two libraries here one is called numpy and another one is matplotlib. So I will write here numpy and matplotlib and run it. If you hold shift and press enter it will run the code and take it to the next line. So now what we will do we will just we will have an example here of 1D Fourier transform okay.

So I can delete this code and add a code here. So first we will define as we said we will use a user defined EFT function okay here we will define a DFT function. So now here if you see 1D DFT function is n which whatever the signal length we will give okay then n is we will define a length n means a variable n for which the DFT function will iterate for or will go from one to another and if you see this one this is exponential matrix creation and this is the DFT creation which is E dot product of E and X okay. So this is what our DFT definition is so we will just run this definition and go to next line and what we will do we will take few add define few signals and add it together to see what is the frequency we are getting in it okay. So if we see here what

I have taken I have taken SR sample rate of 100 so you can say 100 hertz then the time will be 1 by SR the sample rate or we can say it here FS which is sampling frequency we can say FS is the best way to use it so FS so now if we enter and then we took few sign waves of different frequencies here okay so in one the frequency is 1 hertz and second is 4 hertz and 7 hertz so if we see how our curve looks like it will be like this see okay so this command plt dot figure sets the figure here of 8, 6 size okay so if you see the proportion is 8 is to 6 okay now plt T means X what I should plot in X that should be T and what we should plot in Y that is X1 which is the amplitude of the signal added signal okay so this is the short hand notation to add into this means this thing means X1 equals to X1 plus this quantity here okay so this means this now if we if I want to put a label so here I am putting label then if I want to put label for your X what I can do here I can have say time and run it again okay so here is this is the signal we have now if we now what I will do we will do a Fourier transform of it using the DFT function we defined also the using the fast Fourier transfer function of NumPy library okay so just we have we will write this code here okay so here X1 using DFT function it will call the DFT function provide with the values whatever we are giving as X1 this X1 value okay the signal final signal which is created we will give it into the DFT function and the DFT function will calculate the DFT Fourier transform and provide it into X1 capital X1 variable now we will use the sorry using FFT defined in NumPy library okay so NumPy library we will use so NumPy has a class called FFT or the object called FFT inside which there is a FFT function okay so you don't need to understand this but see say NumPy is a very big library in that there is a short library called short library called your short library called FFT and inside which you have actual FFT function okay so you can understand it by like this so inside which you can give your signal and it will calculate the FFT of it now this function generally you can say the general format is np.

fft.fft into sec null comma number of points say you have a very large signal but you have to calculate for first the FFT of first hundred samples so you will give number of points there so it will calculate the FFT for the first hundred samples so we will also see that in the example in upcoming examples so here we will change this to fs fs fs now if you see here if I run this you will see what FFT spectrum okay like this say you have near one one Hertz one signal then positive side and negative side and like sine wave have both side now but you see there is lots of folding I have done here means folding means assigning different matrix like different way we have done here okay so like the first half is the the second half is the positive frequency and the third half is the negative the first half is the negative frequency okay we have done kind of folding here and we use something called FFT shift to fold it so what will happen if we don't do that okay let's see in the next example so we will have here we will remove this FFT shift okay now instead of that we will just have we will remove this frequency also frequency thing frequency so this is just a means like this is this is just a x-axis so we know that x-axis is frequency so we won't bother about it much so here it's error because there is nothing called function called P so now see now FFT spectrum is a is not like the ideal spectrum okay or DFT spectrum is not like ideal spectrum you have to fold it okay you have to fold it to make it look like a real spectrum so that's what FFT shift does so here it starts from 0 to say some x value or the whatever the half of the frequency you are using in half of the sampling frequency remember half of the sampling frequency and this also this starts from say minus to here that is there so the first half is your real FFT and the second half you can ignore okay so this is how the FFT is calculated okay so now if we proceed further let us take few example few more examples of 1D FFT like standard functions okay so let's take a simple sine wave and see what is what we will get okay so let's take a simple sine wave here so we have so we will put here another label x label and then time and enter so okay so we have a sine wave high frequency sine wave here okay so now if we try to find out its FFT using our own function and as well as the function provided by the numpy library what we will what we should expect the same thing whatever our function is calculating so if you see here see almost same amplitude and same thing what we expected so if you take if you remember clearly in one of the examples in week 4 we said that the we said that the affourier transform of rectangular function is is a rectangular purse is a sine wave okay but here you will see something interesting here you will see something interesting which is pardon me so it's saying there is no variable called x3 so this is not x3 this is x13 so here also it's saying x3 so it's x13 okay numpy defined function okay now here oh pardon me so this will be x3 and this will also be x3 what I missed here is not calculating x3 using self-defined function and DFT of small x3 sorry this is x34 okay now yeah so self-defined and this if you see here this is very interesting that this is not exactly a sine wave okay this is because the discrete Fourier transform is a complex number and if you calculate the if you calculate the what real absolute value it won't show the sinc function but stay tuned kind of we will see in in the 2d Fourier transform it's actually sinc okay you will see there it's actually sinc okay so let's take a few more examples like you are we already we already discussed that the Gaussian Fourier transform of Gaussian is always a Gaussian let's first let's verify this is it true let's see so here as I said we will use a different technique we will say we will have 1024 we will compute for only first 1024 points so so we define here x4 the signal okay if we see here this the signal length if you see the signal length here len of x4 len of x4 is len stands for length okay so len of x4 x4 is 80 but we are computing FFT in 1024 points now what will happen okay so it will calculate it will take that and it will extend the extend the actual signal with zero padding on the both side okay means kind of it will have a zeros on the both side and it will create this kind of graph so here if you see the zeros is extended towards this thing but the feature is Gaussian okay so so let's move on to another type of function which is called sinc function let's see whether the sinc function is results to rectangular or not okay results to rectangular or not so let's define the same function here so if we define a sinc function and let's try to calculate the DFT so what we are getting is nearly a square wave okay square pulse not wave nearly a square pulse so what is happening here is you have lots of approximate you are discretizing in both time and frequency domain which is making the system overestimated under estimate at the edges okay because of which this shoot up and shoot down and stabilization is coming here okay so this thing can be tuned by having a factor means a factor time dependent factor or the some kind of slope factor into the sink adding the slope factor into the sink you can have a proper square wave structure but this some this kind of approximation happens okay now if you see the if you trace the average line it is just a rectangular pulse okay and then let's let's see a few more example we will take something called sawtooth wave yes you have seen saw right so here we what we took we imported scipy library which is also a type of library in Python in which there is a signal a small library called signal okay and using that we are calling a function called sawtooth function see it's looking like a saw hacksaw so this that's why it's called sawtooth function means one side right time right angle triangular wave which is means like propagating now if you see this if you can imagine you can imagine it like a sine wave also near to sine wave so you can have it as a multiple sine wave correct you can imagine this as a multiple sine wave correct so you you will think of as a having one prominent sine wave and then gradually decreasing means different harmonics of sine you write so let's check whether it's true or not so so let's see here see it has a one prominent frequency and the hormones start gradually decreasing and if you add it up it will make it a sawtooth in time domain okay so remember Fourier series okay now let's move on to the interesting example which is 2d Fourier transform okay so okay so let's see first what we will take we will have two point approaches a particular distance apart okay so let's define a grid first okay let's define a grid first the grid will be say we will have 512 cross 512 we will say 512 cross 512 meters we have a grid okay this have a area 512 meter by 512 meter where we will put screen or a screen say or you can say antenna or whatever sensor at say 200 comma 250 means in the center of the means this is row versus column so row means your y-axis and columns means your x-axis so your row that your y-axis is 200 and your x-axis is 250 and the another one is at the center which is 250 cross 250 okay near about the center not the center because it's 512 so the center will be around 256 okay so we will initialize this parameter and we will create we will create a inverse Fourier transform screen sorry okay okay so what we have here we we have two point source which we are doing inverse Fourier transform okay we have two point source we are doing inverse Fourier transform to get the electric field distribution on the aperture or the we can say instead of doing IFFT we can do IFFT also let's first see what's what we are getting okay so now we are getting the inverse Fourier transform screen as a fringe pattern means like fringe pattern or the if you remember the Young's double slit experiment you have bright and dark fringe say if I remove this I we have affected to okay so what will happen you will get the same because inverse Fourier transform of two point source or the Fourier transform of two point source or two point source I am saying or rather than source two point aperture in either do you inverse Fourier transform or Fourier transform will get this kind of pattern here okay because those are Fourier pairs okay if you have if you if you are doing in doing IFFT in one domain you will get in another domain the fringe pattern type okay so similarly what we will do we will we will see if we have a thin wire screen okay if we have a thin wire screen or thin wire aperture what will happen so we are defining a thin wire here and let's see how the thin thin wire aperture looks like so let's do an FFT over here and after that let's write our plot routine so one thing which I missed here is this here we are creating subplots two plots and we are using IM show this is a means type of plotting where you plot it as a what as an image okay so here if you click you can see here there is a thin wire because of which we are getting this kind of diffraction pattern okay now let's see if we have a rectangular aperture this is the interesting thing which which I wanted to show because we discussed that the this kind of aperture will result into sinc wave okay now if we have a rectangular aperture what so if we have a rectangular aperture you should you want you will see a sinc wave right you want to see a sinc wave here so see if we have a rectangular aperture what you are seeing there is a gradual decrease in the pattern so this is kind of same okay as it as the dimension increases or the time increases the effect of the fringe decreases fringe or say the intensity of the distribution decreases the center intensity is the most now why it is like plus sign because it's not circular it's a rectangular surface so it has abrupt ending at the at two corners so the diffraction pattern will be more towards the hedge not the corners okay now if you see here it is shifted by some function some it's not in the center it's in the side of the screen but if you remember clearly if you shift a function in one domain there is only one phase term multiplied with the Fourier transform quantity which is a constant value so if you're plotting an absolute value you won't be able to see that shift or the scaling factor now let's move on to say if we have a circular pattern instead of circular aperture or instead of a rectangular aperture so what we should expect we will see what we will expect like sinc function gradually decreasing over the area so if you see here see the sinc function gradually decreasing over the entire thing but if you see clearly here what you will see here is there is blobs this is because this is not a perfect circle because every grid is like a pixel pixel is a small box means rectangle square box so that's why this is there is a blob blobbing nature is coming here due to the discretization of the screen okay screen or say aperture okay now let's if we have two circular aperture nearby separated apart okay two circular aperture with a separation what will happen okay this is interesting here we will step into the realm of interferometer interferometric principles so if we have two circular aperture with separation say 50 meters we will see some kind of resolution decrease see here we have resolution decrease say if I increase this 250 200 so what we will see see we got more resolution means the the separation narrowed means that if you see here the first fringe or the first kind of lobe or the first distributions width is narrower than the previous one okay so if I similarly if I reduce it to 10 what I will get I will get a broader pattern than the previous one see here we have a broader pattern okay because it's almost collapsed with each other okay so we have an elongated pattern because it became kind of a ellipse structure so we have an elliptical pattern so radius is 20 that's why so if we have this as 20 just near around in the border so we have kind of different different pattern so this is say we have like a ring kind of aperture then what will happen so if we have a crazy crazy aperture like a ring pattern what we will see is the following okay so here we will see the main beam at the center whereas we have a different kind of harmonic ripples propagating out different not different kind of harmonic ripples but it has a different relation if you see here it is equally spaced every rings is equally spaced here it's not equally spaced okay it's equal at a certain length but there are two kind of rings which is propagating okay inner and outer inner and outer okay now let's move to another topic which is correlation okay let's first define two signal okay let's define let us define two signals one is say we will have one sync signal and one kind of rectangular pulse so say we have this two now what we will do we will correlate this we will correlate this okay we will correlate this so what we will get for signal one we got the same signal as the sink but with an increased amplitude okay whereas we got triangular for the first one and if we combine both we also then also we get a triangular only okay because what is happening you are getting the maximum value when the sink is within in is in the center and again the gradually gradually the value is decreasing but if you see the width width is almost same or this width is different than this width okay there will be might slight difference which you can see the eye naked eye so let's do another thing we will have a Fourier transform and see how the spectrum looks like okay so here we will do FFT of both signal and the correlation of the both signal what we will get says for sink we got a rectangular pulse for rectangular pulse we got a sink absolute values of sink and if we combine only also here it's R XX of signal 1 and here is R XX of signal 2 that table means that there is a mistake so whatever it is now if you see here you will see if we have if we correlate signal by itself the six figure we will get we got a means amplified sink which also results in amplified it rectangular pulse and if you see we have a triangular wave here we are getting kind of a triangular function of some kind of Gaussian or say decreasing sign some of signs okay as we discussed for Sautu okay so now let's see what happens if we do something called filtering okay so if you remember the concept from the heart heterodyning principle what we do we filter out the signal but let's before that just let's just verify something called why minor kitchen theorem so here what we will see we what we did why not suggest that if you want to calculate the cost relation what you may have to do you have to calculate the FFT of the first function and multiply it with the conjugate of the other function if you do that what you will have is this one so if you see the both functions each both of them looks almost same so they're using by using vinyl kitchen theorem has lots of means approximations involved or losses also there but it's almost same and it's very computationally very fast so this is that now let's understand something called filtering say we have multiple signals which is combined together okay let's see we are if we have say three signals which one is one at 30 Hertz which is an a cell three signals which has bandwidth of 30 Hertz and centered at four frequency 3 kilohertz 5 kilohertz and 9 kilohertz okay so here we will have few signals called FM which is how which has 30 mega 30 30 Hertz bandwidth it okay and we will have and we will generate one two three three signal okay and see let's see we have three message signals of message signal we are seeing is this is a message signal or some some sort of signal okay now we what we will do in real in the real time what happens what happens the signal is modulated okay is modulated at higher frequency so say in heterodyne receiver what what we were discussing is we have something called h1 line which is at 1.4 gigahertz which we want which we want to detect but our systems work at much lower frequency than that so what we do we use a heterodyne principle multiplied by some signal and bring it down so similarly what we will do we will multiply it by a higher value signal and we will create something called frequency modulated signal okay so not amplitude modulated signal by multiplying that to this so what we are getting we are getting the final signal as this much this is kind of garbage you can't understand what is what is what okay so what we will do we want to separate out say the a2 signal out of it so we will calculate a Fourier transform of it so here you will see the essence why Fourier transform is so important okay see here what you are getting you are getting distinct feature at points okay means although it is so jumbled up but you are you are getting this kind of distinct feature okay now what what we will do we will filter we will create a filter at the center frequency of 5 so the 5 kilohertz band okay so we will create a filter of say 50 plus minus 50 Hertz okay so since it's a very narrow filter you will see like a square but it's very short so filtering is nothing but what you will do you will multiply this by the Fourier spectrum we got earlier which is this one so if we multiply this what we will get we will get this okay now if we do the inverse Fourier transform of this we should get the actual signal okay so so what was the what what we gave is this signal what was the original signal at 5 kilohertz band is this and what what we recovered is the same okay so you can recover the signal this shows the importance of filtering now if we know the importance of filtering and Fourier transform then there is something called sampling okay then there is there is something called sampling and the loss factors related to that let's quickly go through what is sampling okay so sampling is nothing but changing a continuous changing a continuous time signal or that continuous space signal or the continuous signal into discrete form by sampling at particular weight okay at particular interval you will sample so if you want to recover the signal intact the sampling rate should be greater than 2 FM where FM is the your bandwidth signal bandwidth you can say okay so you have say signal emitting at means having a bandwidth of 10 megahertz so twice of that you should sample why this because using heterodyne principle you can shift the frequency center frequency say you have 10 megahertz across 1.42 gigahertz but using heterodyne principle what you can do you can shift this to say 0 to 50 megahertz okay so then what you have to do you have to sample with 100 megahertz twice of this how much whatever your start limit and end limit is with twice of that you need to keep your sampling frequency okay we will see in a bit why it is so important this is important because that will cause aliasing what is aliasing okay let's discuss this this was discussed in week 4 also but we will discuss a bit more about it so if we have say a bandwidth signal which has a bandwidth of say minus omega say of say 0 to 10 megahertz 10 M okay now if you are sampling at the rate of what say 50 megahertz okay what you will have you will have a frequency spectrum in discrete Fourier transform I am talking about discrete Fourier transform okay so it will go like 0 to this okay and here you will have this and here to here okay now say instead of 10 to this we have 1 to something so this will have this won't be like this this will be like here and here okay now what will happen what will happen if I decrease this to 20 it will be same you will get same kind of pattern okay but if you decrease it further you will see a overlapping frequency this is very difficult to understand in frequency domain so what we will do we will take it into time domain and see how and why this is important there is something called over sampling over sampling means you have very you have leftover band between two consecutive bands okay so when you do over sampling you have leftover remaining place here okay which is not in use so perfect sampling results in no space but it is thumb rule that you should leave some guard band over here okay so such that if there is some kind of instrument error or drift it won't go under sampling and you have a signal loss so it's like a kind of thumb rule that if you have a kind of bandwidth a bandwidth signal then you should have the sampling frequency or at least three times the bandwidth Fs should be three times the bandwidth so that you will have 1 BW of guard band over here so this will be a 1 BW of guard band over here okay so this is what we use in we try to follow in radio astronomy now let's see what understand bit more looking into time domain of it so let's move on to our notebook so let us first let us first quickly again discuss what is aliasing means in time domain okay now here by definition what we will have is sampling theorem says that you should have more than twice of FN okay now now this is why because a continuous time signal if it we have a continuous time signal okay if we sample by the Nyquist rate this is what also called Nyquist rate it we can recover the signal back okay kind of more kind of intact at such that we the no information is lost if this doesn't satisfy what will have we will have some other frequencies coming which is kind of which is the signal will become indistinguishable just we won't be able to distinguish like which signal it is and there therefore the information is lost so let us have let us see this example over here so we have a signal at at 2 kilohertz okay this is a 2 kilohertz signal okay going means we have 2 kilohertz signal sampled at what frequency sampled at at 18 kilohertz frequency okay so we have 2 kilohertz signal sample at 18 kilohertz frequency so we will get this much amount of sample okay so 2 kilohertz and 18 kilohertz frequency means in one cycle you will get around 9 samples okay so 1 2 3 4 5 6 7 8 9 so in one cycle we have 9 samples okay so 5 and 4 9 so one cycle 9 9 samples so let's have another signal let's have another signal okay which has a frequency of 10 kilohertz but the sampling rate is same as the previous one okay so what the sample is sampling is happening at different frequencies which you don't want means like see the sampling is like kind of haphazard so what does this mean actually if we see the plots here here the here what is happening the sampling rate is much much lower than the Nyquist rate okay so if we plot everything together all the three all the plots together what this is what we will see see this was a 2 kilohertz signal and this was the 10 kilohertz signal if we sampled at Nyquist rate we were able to recover almost the same signal see almost the same signal there was there are some losses but almost the same signal but if we under sampled it the signal is entirely lost if we over sample it no issue you will get back the signal but the extra bandwidth is utilized is which is not so good for the system or it might means you might over use your your storage space which you which is not required if you properly sample it so this is what what happens if you do if you go under sampling okay so with this with this I will say thank you