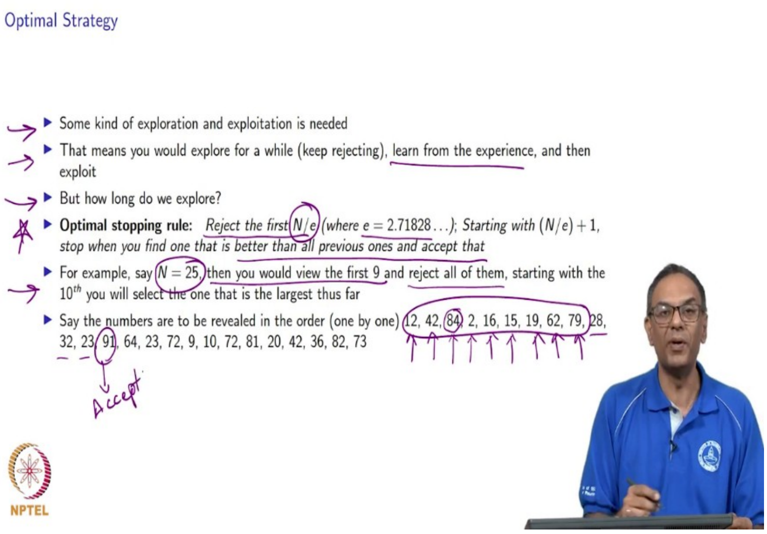
Decision Making Under Uncertainty
Prof. Natarajan Gautam
Department of Industrial and Systems Engineering
Texas A&M University, USA

Lecture – 11
Solving the Secretary Problem

Now, we will see how to solve the secretary problem or the marriage problem or the special problem where we pick are the best among the n pieces of paper.

(Refer Slide Time: 00:28)



So, I am sure many of you would have thought about the best way to do this and it is very common that people think of some way of exploring first and then exploiting. Now, we will use these words “exploration” and “exploitation” a couple more times during this course. The idea is that you will first explore and see how the numbers look like; you will keep rejecting them just because you want to get some samples.

Let’s say you are someone who has never hired a secretary in their life or think about how do I go about when I have no idea who would be a good person to have. So, you will explore a little bit; you will try to see how the lay of the land is and learn from experience; then you would exploit. Exploit sounds like a bad word. Basically, what we mean by that is, once you learn something, you will say, “I will come up with some rule and then I will adopt it; I will make sure that I try to get the best secretary possible.” So, that type of a strategy where you would reject a few people for a while and then, once you find a good person, you would hire them, is a strategy that a lot of people would come up with for this.

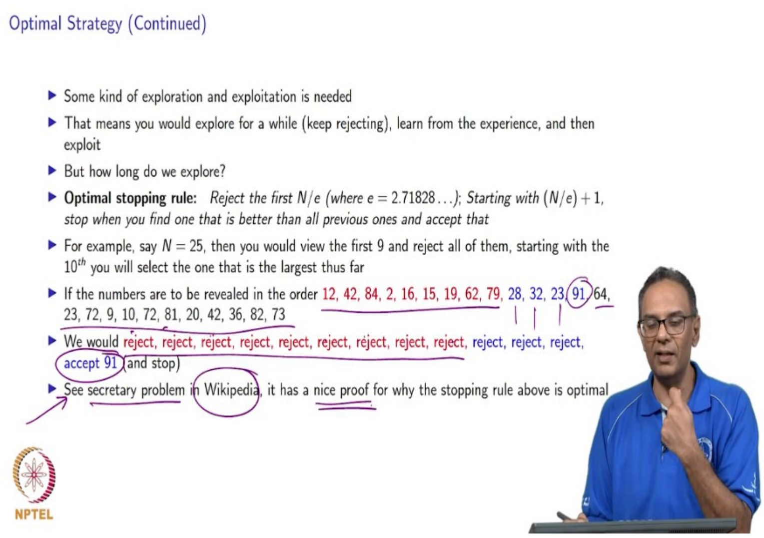
But how long do we explore? Now, that's a question that I don't know if you would have thought right off the bat. You do not want to go on exploring; out of the N, the first few, you want to explore and then for the remainder, you want to exploit.

So, the optimal stopping rule which we will discuss, is the optimal policy among all the policies when N is large. So, what you would do is, you would reject the first N/e. So, N is the number of papers; e is the usual natural logarithm's base. So, e is 2.71828 approximately. So, you would reject the first N/e. Of course, N/e may not be a round number. So, you would round it down and reject all of them. Starting with the next one, you will say, "Okay, we will look at this to see if this is better than everything". We have seen so far that if that's the case, you would accept it. If it is not, you will reject that and go to the next one and then see if that is better than everything you have seen that far. If it is, you will pick that. Otherwise, you will reject it and you keep going this way. Essentially, you will see the one that is better than all the previous ones and accept it. So, that is your stopping rule. You will stop when you found one that is the largest thus far.

This policy like I said before, is not guaranteed to give you the optimal solution because what if the largest one shows up in the first N/e. That's it; then, there is no way for you to select it. It could also happen that the first N/e is such that you did not get too many of the large numbers and then, the very first one that was larger than those you pick; but there may be others that are coming in the future. So, we don't know that. So, it's not guaranteed that you would do it. But among all the policies that you could think of, this one is going to be the best.

So, let us do a little example. Let us say N is 25; now, 25 is not tremendously large; however, it is useful for the purposes of the proof of the optimal policy. However, it is good enough actually for illustrations and also, when you put it on a screen, it is easy to see some of this. So, what you would do is, you would reject the first 9. So, 9 is now 25/e; that's the number that is the nearest to that. We will reject the first 9 of them. Then, starting the 10^{th}, you will look and see which is the best we have seen thus far and go ahead and pick it. So, think of the following. Pretend like these numbers are revealed one by one. First, 12 is revealed, we open that up; next, the 42 is revealed; then 84, then 2, then 16, then 15, then 19, 62, 79, 28. So, you will automatically reject all these because you want to see how the numbers look like. Remember that you have not seen the remaining of 25-9, which is 16. So, you have not seen the remaining 16. You have only seen the first 9; you will reject them all. Now, you will see, "Well, if I find a number that is larger than 84, I will accept the first one that is so." So, you look at 28, you will reject that, 32 reject, 23 reject and then, you will see 91 and you will go ahead and accept. And, I am going to change to the next slide where we have kind of explained this.

(Refer Slide Time: 05:13)

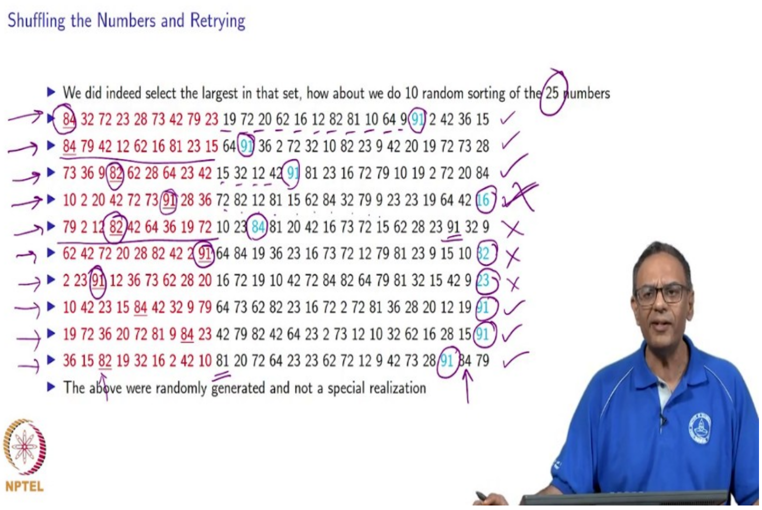


Notice that you will reject the first N/e. So, that is reject, reject, reject, reject, reject, reject, reject, reject and reject. So, that is the red color rejects of the first N/e which is 9; then you will reject 28 because it is smaller than 84, reject 32 as it is smaller than 84, reject 23 as it is smaller than 84 and then you come to 91, and you will go ahead and accept it. At this point, you would say "stop". Now, turns out in this case we got lucky. If you notice, 91 is the largest.

So, we did really well; this rule really helped us. I promise you this example was one in which this was the first randomly selected numbers that I threw in and it turned out that I was lucky enough that just worked out for me. I am going to show you next when this could kind of mess up in some situations which we will see in the next slide; however, I do want to pause for a moment to let you know that this problem, why this is optimal is nicely derived in a Wikipedia site.

Now, I want to pause for a moment to let you know that in general, Wikipedia is written not sure of the whole Wikipedia, but articles in the Wikipedia are written by people who does not have to even be an expert, but who are someone who knows the topic somewhat well and has written something up. Very nice of them, but it is not always that a Wikipedia article is going to be absolutely accurate. Therefore, you want to be a little bit careful; however, from what I looked at and for this article, if you searched for secretary problem in Wikipedia, you will actually find this and you will find a very nice proof. And, I do not want to go over the proof for purposes of this course; however, I would recommend you to go ahead and look at the proof if you have the opportunity and see why the stopping rule that we talked about, that is rejecting the first N/e of them, and then after that, looking for the biggest number and accepting at that point, is the optimal solution.

(Refer Slide Time: 07:28)



What if we had actually gotten the same 25 numbers in some other order? We shuffled them and it came in some other order. Let's look at the first set. So, let's say you first got 84, then 32, then 72 and so on. So, clearly the largest is underlined and you would perhaps say at this point, among the first 9, my biggest number is 84 and therefore, anything that is larger than 84 say reject, reject, reject, reject, reject, reject, reject, reject, reject, reject and then reject, and I will see my 91 and I accept it.

Again, we have been lucky that we actually picked the biggest number here. So, 91 is the largest I am picking. I am shuffling the same example up just to show you that there is no guarantee that this was the order in which we got the numbers because there are 25! ways that I could have actually selected these numbers. So, there is no guarantee that 91 actually is the one that we are going to pick. But somehow, we got really lucky in the first three. So, what I did was, I wrote a little program and asked it to shuffle these numbers up. So, turns out in the second example as well, we again have this 84 in this first set of 9 and therefore, anytime I see a number larger than 84 which happens to be 91, we picked that and we have done well.

Now, let's look at the third one. Again, this time it is not 84, but its 82 that is the largest and the third set. And therefore, anytime I get a number larger than 82, I say let's do it and turns out the first four are not and then, I come to 91 and then, I go ahead and say yes. So, we have been lucky in the first 3.

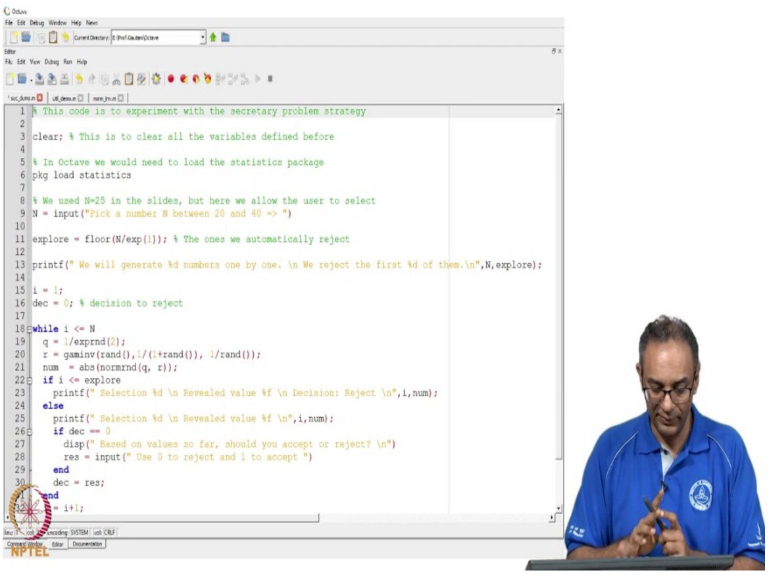
Let's look at this example 4. What happens is our friend 91 is in the first N/e, the first 9 itself has it. So, there is no chance for you to pick the best one. So, turns out that none of the other numbers are actually going to beat 91; we know that and therefore, you would pick the very last one. Well, according to our rule, you will reject that as well and you will be without a secretary. But on the other hand, you know what one would do is, go till the very last one and select the last one.

Now, you could have been cleverer and say, "I have a different policy; every time I get a number that is even somewhat close to the largest, I will pick that." You could do those types of things and it is not clear, it is not easy to prove that is optimal. But you know it is possible that it's a reasonable strategy. Now, let's look at another example. In this case, what happens is the first 9 of them has 82 as the largest and we say every time we see a number that is larger than 82, we will say that is it. So now, what happens is that our friend 84 that was the best in the first couple of examples shows up before the 91 and we look at it and then go ahead and pick the 84, and we miss out on the 91 that is coming much later. Now, this is a reasonably good solution anyway; however, that is one of the downsides because you do miss it a few times.

Now let us look at the next example. Here again, the 91 has showed up. Every time 91 shows up in the first 9, you are never going to beat that 91. And therefore, you will go till the last number and pick that. So, that is not going to work out. Again, the next example also has a 91 in the beginning and therefore, you will not pick any number that is larger and therefore, you will be settling down with the very last number that you see.

Now interesting enough, look at the next three examples, and again, I did not make this up, I actually randomly generated. If you look at the next two of them, 91 ended up being the last one and I ended up picking it. And, in the third one, 91 came before and you ended up picking it because fortunately my 84 came later. My 82 was stuck here in the first N/e and therefore, I got lucky; otherwise, I could have picked 81 as a matter of fact. Any different things could have happened and we got lucky. So, this is the last slide and what I am going to do is to do a little demo. Let us try to do this in Octave.

(Refer Slide Time: 11:30)



So, what I am going to do here is, I have written up a program and I would share this program with you. So, be sure that you go to the website and download this program and I would encourage you to play with these numbers a little bit, but I want to go over the program a little bit.

So, this is a program or a code that is going to help us solve the secretary problem. So, first step like always, is to "clear" so that you do not have variables that were there from before. Once again, we load the statistics package. By once again, what I mean is from the previous lectures. Then in our slides, we use N=25, but here I am going to let the user pick a number.

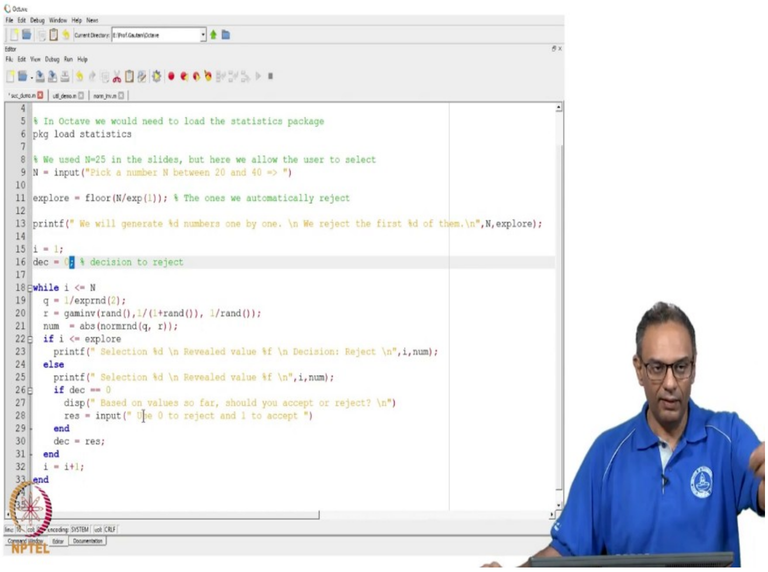
So, this also is a little user GUI if you will, not graphical, but at least the user interface. Let's say I let you pick a number between 20 and 40. If it is larger than 40, it gets a little bit cumbersome. Of course, it will work. You are welcome to put 10000 if you want to, when you try it. But, let 's say you put between 20 and 40. Now, by explore remember you are first exploring the first N/e of them. So, exp(1) is e^1 which is e. For N/e, I take the floor of the function, $\frac{N}{e^1}$ because I want an integer. Therefore, I take the floor of that and I would go ahead and explore; that's my value. So, whatever number I take as N, I divide it by e and that gives me how many I want to explore. In case of 25, explore will be equal to 9.

Then, my program will reject the first explore of them and then, we will see what we need to do. So, this is what we would do. I have a little index i, that goes from 1 through all the way to N because I keep revealing the numbers as they go; we will keep rejecting the first N/e of them and then, we will give you the choice - do you want keep or do you want to reject?

So, the key in this is, I have really diabolically generated some numbers. Let us look at this. I first generate a random number q. This q is one divided by an exponential. So, remember that exponential distribution comes up with some really small numbers from time to time. At times, it could be larger than 1. So, the mean is 2; sometimes, you get a number smaller than one; sometimes, you get a number way larger than 1 and we do not know what it would be like. So, that will be my q.

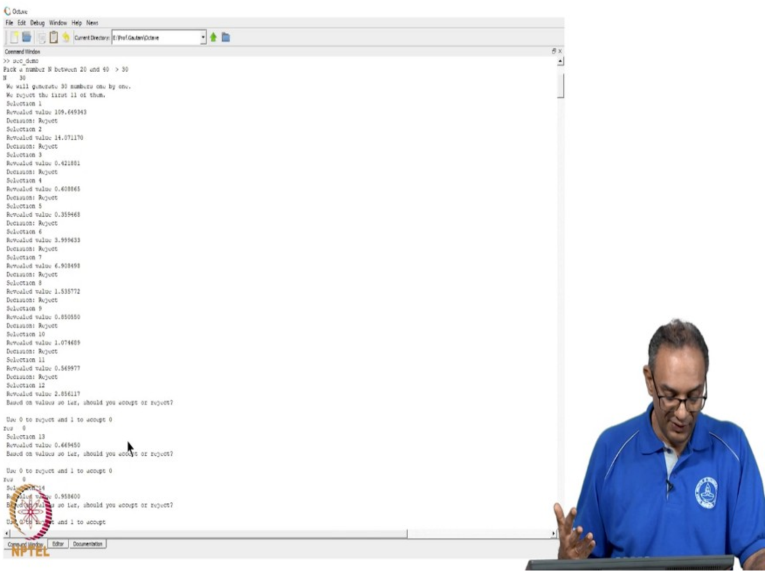
Then, I have an r which would be the inverse of a gamma distribution where I randomly pick an inverse of the CDF and I get another number. So, I have two random numbers which I use as my parameters of a normal distribution. q is going to be the mean parameter and r is going to be the standard deviation parameter and I get the random number and I put it into Octave and create a normally distributed random number; take its absolute value because normal can be positive or negative. So, I take the absolute value. So, that is my number that gets generated each and every time. When we see the examples, you see these numbers could go all over the place. I have really picked some nasty ones. If I pick some nice distributions actually, you will see we do pretty well. This would be pretty nasty.

(Refer Slide Time: 15:00)



And, I do not know how many times you are going to do well. So, what we do is, we will explore for the first N/e, then once that is done – hence, else, what we will do is we will say my decision is 0. That means, I will reject. If I have been rejecting, till I reject, I will keep releasing numbers and ask you the question - do you want to keep this or do you want to reject? And, once you have accepted, then what it will do is, it will reveal the remaining numbers. It will just reveal the remaining numbers to see if you actually have picked the largest one or not. Let us play this. It is an interesting game that we play and then, you keep doing this till you get all the N.

(Refer Slide Time: 15:38)

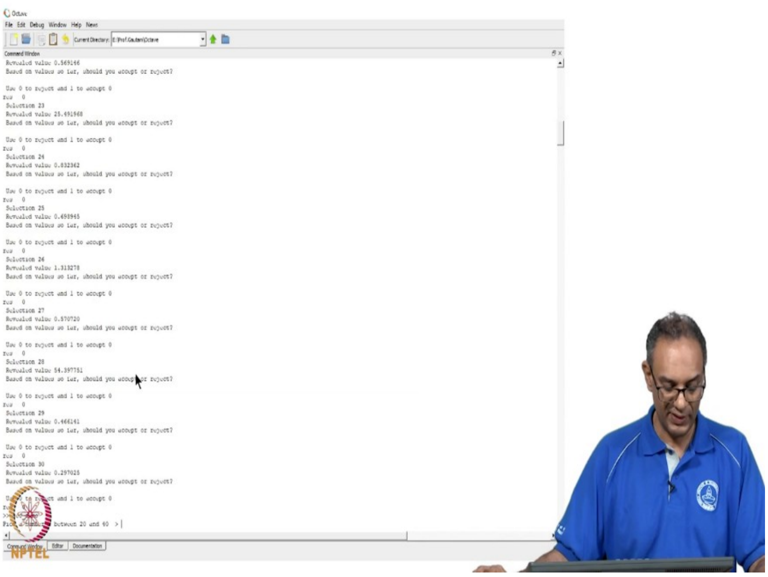


So, the name of this program is called sec_demo. So, we do sec_demo. I understand that the font size here is pretty small and you have to really squint your eyes to read. If this is difficult, I am going to talk through this. You are welcome to download the program and exactly type what I am typing and see on your screen how this is working out. So, I am going to hit sec_demo and hit enter. As we expect, the first question that it asks me is to pick a number N between 20 and 40. So, let us say I pick 30. So, I am picking N=30 and I hit enter. Then, what it does is, it tells me that the first 11. So, I am going to reject the first 11; so, it will generate 30 random numbers, but one by one, it will reject the first 11, that is N/e, the lower bound of N over e, and it will reveal some values.

So, the first number it reveals is 109.64 or 65, the next number is 14, the next number is 0.42, 0.6, 0.35, 3.99, 6.9, 1.5, 0.85, 1.07 and this last number is 0.56. So, these are the first few that it rejected. If we notice, the decision is to reject. So, clearly the largest one is a very first one we have seen.

So, anytime I get a number that is greater than 109.65, I will say, let's use that. Clearly, the first number that we have is not larger; so, we are going to reject; so, I am going to put 0 and reject. The next number is 0.67; again, we will reject; so, I will do 0. Now, when you run your program because your random numbers could be different than mine, you might get a different set of numbers. So, do not be surprised when you do not see the same set of numbers. That is because it is randomly generating.

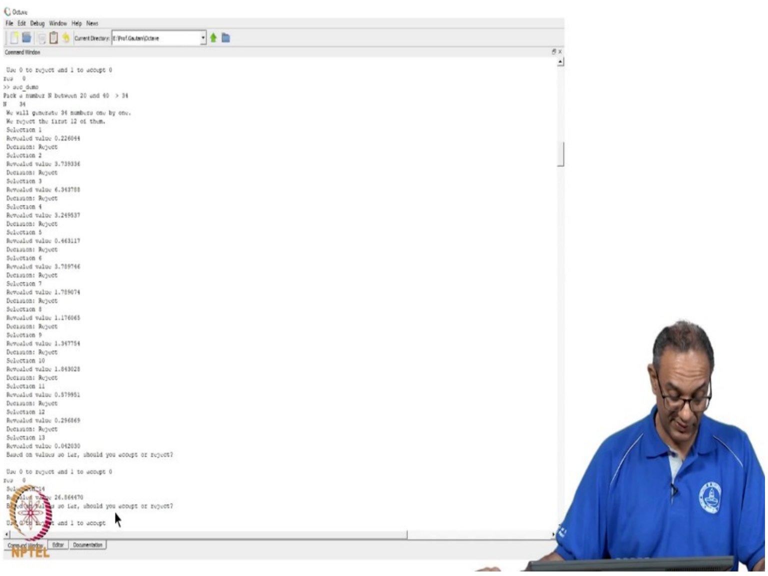
(Refer Slide Time: 17:47)



Now, the next number is 0.95; clearly, we will reject that because we need a number almost as large as 110. We will reject 1.40 as well. Finally, we get a large number 57, but still not as close as 109; so, we will reject; 5.8 we will reject, 0.50 we will reject, 7.82 is slightly larger but still we will reject, 0.2 is too small and we will reject, 0.5 is too small and will reject, 0.56 is too small and will reject, 25 is better but still we reject, reject, reject, reject.

In real life, you would say we will just take the 54 because we do not seem to be getting good numbers. In real life, you will accept this, but I am going to go ahead and reject it because that is what our rule says. I will reject it and I will reject. Basically, you would end up accepting, but I am going to press 0 just because I think it does not say what to do with the very end. So, let's try this one more time. Let us do sec_demo second time. Now, let us pick a number between 20 and 40, say 34.

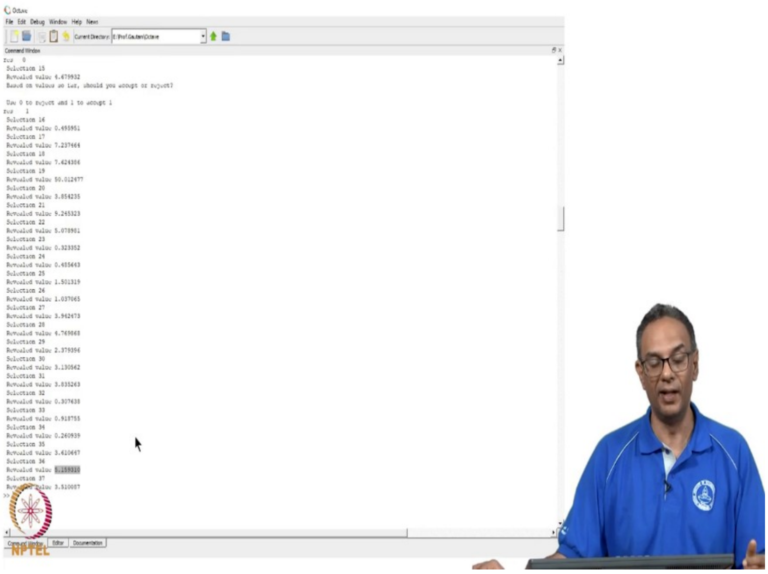
(Refer Slide Time: 18:59)



Now, what happens is, it picks the first 12 and rejects them all. If you look at, the largest number that we have here is 6.34. So, you look at 0.04 and you will reject. And then, the next number is 26; so, it is bigger than your 6. So, this 26 is larger than the biggest number we have seen so far which is 6. So, I will accept this. But after I accept, what will happen is it will tell me what is my 15th number through my 34th number and see if my 26 was the best. So, turns out that the 15th number is 1 point; so, you look at all these numbers, then 0 point, 8 point, 2, 4, 2. So, we ended up being lucky this time and we actually did select the best number.

I should stop on this wonderful note, but let me just be a little bit masochistic and see what happens again. Let us try a third time, say we pick 37 as the number. So, we will reject the first 13 and so, if you look at it all the numbers, they are somewhat small except this 3.99. So, this is almost 4; that is the largest number I see. So, you look at 1.56 and you will reject. So, if you get a number larger than 4, you will accept right away. You got one and I will accept this.

(Refer Slide Time: 20:22)



And then, I hit enter. Turns out that if I was a little bit more patient, I will hit this 50; that would have been my best choice, but I settled in for a number that was not the best. So, it is not always that you will get the right answer or you will get the best one, but you tried.

So, the policy says I will reject the first few and I take the next number that is going to be the largest that I have seen thus far. Turns out in this previous case about almost 4 was the largest number. Any time, any number I see bigger than 4, I am going to accept. When I saw that, I accepted it and turns out soon after that I got a ton of numbers that were larger than 4 - 7, 7, 50, 9 point and so on, 4.7, you have a bunch of numbers, but 5 point here one more. So, we have about 5 numbers that are larger with one of them being really large. So, we missed getting those. So, turns out that like I said, this is not guaranteed to give you the largest number, but this is a policy that will help you get the largest number with a high probability.

Thank you.