

Essentials of Data Science with R Software-1

Professor Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology, Kanpur

Lecture 21

Computation of Probability using R

Hello friends. Welcome to the course Essentials of Data Science with R software 1, in which we are trying to understand the basic concepts of probability theory and statistical inference. So, now you can recall that in the last couple of lectures we have learned different types of probabilities, simple probabilities, conditional probabilities, based probabilities etc. And we have learned different types of results and concepts like as theorems of total probability, additive probabilities, independence, etc.

Now, my question is very different, in this lecture. You have learnt from the theory point of view that how are you going to compute those probabilities. But in the case of data science, you will have only the data, data and data before you. It will not be possible for you to compute these probabilities manually. There can be different types of variable, large number of variables, large number of data sets, huge number of values.

So, now you have to think somehow that how one can compute these probabilities through the software. Now, as far as I know, I do not know any software which can compute the probabilities directly. If you want to compute some Bayesian probability or conditional probability, at least, I do not know any software which can do it directly with the click of a click of a command.

But you have to think and then particularly when you are trying to work in data science, that conditions under which you have to find out the probability they can be complex. And finding out the exact expression, mathematical expression might be difficult. So, in those cases, if you want to compute the probability, one good option is that you can compute them. In that case, I will say okay, I am not bothered what is the mathematical expression, which is computing the probability but I am more interested in getting the value of the probability.

So, from this point of view, if you try to see or try to observe the types of probabilities, what you have done up to now, what are the basic ingredients? I say, you have learned the simple probability. Simple probability also you have understood what is the connection between the relative frequency and the simple probability. Then you have learnt conditional probability. And after that, you have learnt the Bayes probability.

But if you try to see conditional probability is also a function of simple probabilities. And Bayes theorem is also a function of simple probabilities and conditional probabilities. So now, if you can understand or you can give a logic that how one can compute the simple probability and conditional probability, possibly I am done. Then, either I have one variable, two variable or 100 variables, how does it make any difference? That depends only on the capability of the programmer.

So, in this lecture, exactly, this is what I am going to do. I have taken a very simple example. And I will be taking only a very small number of observations which are simulated from the software, using that data set, my objective is to explain you how you can compute these probabilities. Well, this is only one possible way. Depending on the logic, there can be different other ways. There is no question that this way is correct or that way is correct as well, as long as you are trying to compute the probabilities correctly, all the approaches are correct.

In programming, there cannot be a single approach by which you can always say that this approach is the best approach. So, with this objective, I would like you to watch this video and try to understand what I am trying to convey. Mathematical expression will be zero in this lecture, but we have to see that how we are going to compute different types of probabilities.

(Refer Slide Time: 05:04)

A slide titled "Computation of Probability using R:" with three bullet points. The first bullet point is "Various approaches can be employed for the computations of probabilities in terms of approximate relative frequencies." The second bullet point is "The logical operator can be used for finding different the relative frequencies." The third bullet point is "A function can be written to compute different combinations of these probabilities." The slide has a black border and a small number '2' in the bottom right corner.

Computation of Probability using R:

- Various approaches can be employed for the computations of probabilities in terms of approximate relative frequencies.
- The logical operator can be used for finding different the relative frequencies.
- A function can be written to compute different combinations of these probabilities.

So, let us try to begin our lecture and try to understand this concept from a computational point of view. So, now, there can be various approaches can, which can be applied or which can be employed for the computation of various probabilities in terms of relative frequencies.

Now, why? Because, you see, when you are trying to compute a probability, the probability will usually be computed in terms of relative frequencies solely. And that was the reason that in the beginning I have spent some time in explaining you the relative frequencies and probability and I had tried my best to give a connection between the two.

So, now one option to compute different types of probabilities is that we can employ the concept of logical operators. And if you remember in the beginning, when I was trying to introduce various concepts in R software that are going to be useful, we had talked about the logical operators. So, the logical operators can be used for finding different type of relative frequencies. So, our function can be written to compute different combinations of these probabilities, means a simple programs can be written, which can compute different types of probabilities.

(Refer Slide Time: 06:15)

Computation of Probability using R:

Suppose a fair dice is rolled and its outcome as the number of points on the upper face is recorded as 1, 2, 3, 4, 5, 6

Sample space (Ω) = {1, 2, 3, 4, 5, 6 }

Example:

Suppose we want to compute the probabilities of occurrence of 1's, 2's, 3's, 4's, 5's and 6's.

Note that

$P(\text{Occurrence of 1's}) = \frac{\text{Total number of 1's}}{\text{Total number of trial}} = \text{Relative frequency of 1's}$

So, let me try to take a very simple example that you understand from the theory point of view and I will try to imply the same thing in the R software and I will try to compute different types of probabilities. So, suppose our fair dice is rolled and its outcome as the number of points on the upper surface are recorded as 1, 2, 3, 4, 5 and 6.

So, the sample space is here 1, 2, 3, 4, 5 and 6. And suppose we want to compute the probabilities of occurrence are 1's, 2's, 3's, 4's, 5's and 6's. So, now how to do it, that is a

simple probability. So, probability of occurrence of 1's is simply going to be the total number of 1's divided by the total number of trials. So, this is actually the relative frequency of 1's.

(Refer Slide Time: 07:01)

Computation of Probability using R: $P(A)$

Suppose we repeat the experiment 100 times and the outcomes are recorded and the relative frequencies are obtained as follows:

Example: $P(A)$

A: Event of occurrence of 2's

$P(\text{Occurrence of 2's})$

Total number of 2's = 15

Total number of trials = 100

$P(\text{Occurrence of 2's}) = f(2) = 15/100$

And now, suppose if you repeat the experiment 100 times and these outcomes are recorded, and then the relative frequencies are obtained and from those we can have an idea about the probability. For example, if you try to repeat the experiment 100 times, there will be some number of 1, 2, 3, 4, 5 and 6.

So, suppose A is the event of occurrence of 2's, means the two points on the upper surface of the dice. Then from here we can compute the probability of occurrence of 2's. How? We simply have to compute the total number of 2's, suppose it is coming out to be 15 and the total number of trials are here 100 because we have repeated it 100 times. So, the probability of occurrence of 2's will simply be your relative frequency at 2 which will be equal to here 15 by 100.

(Refer Slide Time: 07:52)

Computation of Probability using R: $P(A)$

This can be demonstrated in R by the sample command by drawing the observations among 1, 2, 3, 4, 5, 6 by simple random sampling with replacement and then finding the relative frequencies.

Earlier we used the table and length commands to compute the relative frequencies.

For illustration, suppose we want repeat the experiment 10 times. This means drawing 10 values and finding the relative frequencies of 1, 2, 3, 4, 5, and 6.

Now, in case if you want to demonstrate this experiment in the R, then we can use the sample command. We have already used it earlier when we were trying to compute the simple probabilities using the relative frequencies. So, the same concept we are going to use here that we will try to draw the numbers 1, 2, 3, 4, 5 and 6 by simple random sampling with replacement and then I will try to find out the relative frequencies.

So, earlier we had used the command table and learned to compute the relative frequencies, means you can recall. So, just for the sake of illustration, we are going to repeat the experiment only for 10 times. Well, I am not repeating 100 times, 1000 times, that you can do because there is a limitation on my slide, means, if I try to make here 100 observation, I will not be able to explain you clearly.

My objective here is to explain you what is really happening and how are you going to execute it through the R software. That is why I can handle here this number of values very easily. So, when I am trying to repeat the experiment 10 times, this means we are trying to draw 10 values and then we are trying to compute the relative frequencies of the numbers 1, 2, 3, 4, 5 and 6.

(Refer Slide Time: 09:06)

Computation of Probability using R: $P(A)$


The command

```
dice10 = sample(c(1,2,3,4,5,6), size=10, replace = T)
```

Population
SRSWR

generates 10 values and stores it in a data vector `dice10`.

```
> dice10 = sample(c(1,2,3,4,5,6), size=10, replace = T)
> dice10
[1] 6 2 1 3 2 2 5 5 4 3
```



So, now, I try to write down here the command `sample` and this is my here population, from here I am going to draw the simple random sample with replacement of size 10 and `replace` is equal to `TRUE`, that means we are going to get here the simple random sample with replacement. So, this is going to generate 10 values and we are trying to store it inside the data vector whose name is `dice10`.

So, dice is repeated 10 times. So, I try to execute this command on the R console and we obtain here this outcome, you can see here this is a screenshot. So, you can believe that as if you are doing it directly on the R console. Now, so you can see here you have obtained these values.

(Refer Slide Time: 09:54)

Computation of Probability using R: $P(A)$

Now we can make use of logical operators to compute various type of probabilities.

Example:

Suppose we want to compute the probability of occurrence of 2's.

We need to count the number of 2's in `dice10` and divide by the length of `dice10`.

This is obtained by the following commands:

Now, what I have to do, we have to compute different types of probabilities or for this I suggest that we can make use of the logical operators to compute various types of probabilities. For example, suppose we want to compute the probability of occurrence of 2's that we already have done, but still, I will try to repeat it for the sake of completeness.

So, in this case what we need to do? We need to count the number of 2's which are coming in the outcome of the values which are stored in the data vector dice10. And we will try to divide it by the total number of observations which is obtained by the by finding out the length of that dice10 vector.

(Refer Slide Time: 10:39)

```

Computation of Probability using R: P(A)
> dice10
[1] 6 2 1 3 2 2 5 5 4 3
> length(dice10[(dice10==2)]) # Counts no. Of 2's
[1] 3
> length(dice10) # Counts total occurrence
[1] 10
Probability of occurrence of 2's
> p2 = length(dice10[(dice10==2)])/length(dice10)
> p2
[1] 0.3

```

So, you can see here that how many values are there here? In dice10, there are 1, 2, 3 values. Now, the question is this how are you going to find out that which of the values in this data vector are exactly equal to 2? Because this number can be 100, 1000 or whatever you want. Here I can show you that you can count it manually.

But for that means, in my limited knowledge of programming, I think, I can use here the logical operators and I can see here try to find out the values which are equal to 2 in the data vector dice10, and I am using here the double equality sign which is logical equality, means it will simply try to check that which are the values which are exactly equal to 2. So, it will give you the answer in terms of TRUE and FALSE.

And then I would like to find out what are those values for which this command is TRUE, that dice10 is double equal to 2. For example, if you see the outcome of dice10 equal to 2, it will come like this for 6 it will say here FALSE, for 2 it will say TRUE, for 1 it will say here

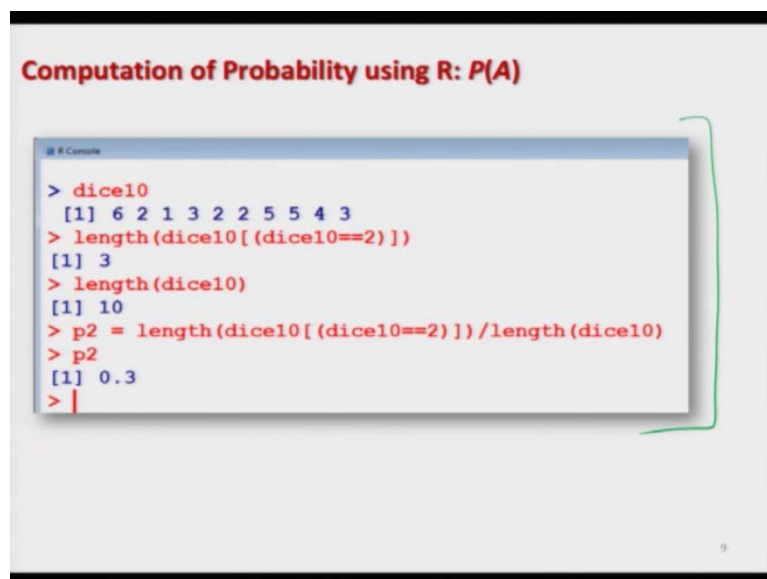
FALSE, for 3 it will say FALSE, for 2 it will say TRUE. And similarly, for this 2 it will say TRUE and so on.

So, it is simply going to count the number of TRUE values or number of TRUE's in this vector and then I am trying to find out the length of the resultant vector. So, this command length of dice10 where dice10 is logically equal to 2 will give you the total number of observations in the data vector, which are exactly equal to 2.

And then I am finding what is length, so, it will give me the total number. So, now, I try to find out here the length of the data vector. Well, this is here 10, but suppose we do not know. So, this command will give you the value here 10.

Now, if you try to find out the value of occurrence of the number 2. I am indicating by here P2. So, this is the length of this dice10 where dice10 is equal to logical equal to 2 divided by the total length of the dice vector. And this will come out to be 0.2 and 0.3. And you can see here this is obvious, there are three values which are 2 divided by 10, which is equal to 0 point 3.

(Refer Slide Time: 13:22)



```
Computation of Probability using R:  $P(A)$   
> dice10  
[1] 6 2 1 3 2 2 5 5 4 3  
> length(dice10[(dice10==2)])  
[1] 3  
> length(dice10)  
[1] 10  
> p2 = length(dice10[(dice10==2)])/length(dice10)  
> p2  
[1] 0.3  
> |
```

So, you can see here you have computed this simple probability very easily and you can see here this is the screenshot, and surely, if you try to repeat it, do you think that are you going to get the same outcome? Certainly not. Well, the probability is very low, because these are the random numbers and the probability is extremely less that this random numbers are going to be repeated again.

(Refer Slide Time: 13:40)

Computation of Probability using R: $P(A \cup B)$

Example: $P(A \cup B)$

Suppose we want to compute the probability of occurrence of 2's or 6's.

A : Event of occurrence of 2's

B : Event of occurrence of 6's

We need to count the number of 2's or 6's in `dice10` and divide by the length of `dice10`.

This is obtained by the following commands:

10

Now, I try to give you here one more concept that how are you going to compute different types of probabilities? They can be probability of union, intersection etc. Once you understand how are you going to compute the probabilities of union, intersection after that you can compute any type of probability. For example, the conditional probability is simply the combination of probability of intersection and a simple probability.

So, now, if you try to see here, I am trying to compute a probability of A union B and I am going to use the same example, same data say that we have just obtained. So, we want to compute here the probability of occurrence of 2's or 6's. So, we define here the event A as event of occurrence of 2's, and B as the event of occurrence of 6. And we need to count the number of 2's or 6's in this data vector `dice10`, and then we have to divide it by the length of the data vector `dice10`. So, this is obtained by the following commands which are based on the logical operators.

(Refer Slide Time: 14:39)

```
Computation of Probability using R:  $P(A \cup B)$ 
> dice10
[1] 6 2 1 3 2 2 5 5 4 3
    T T F F T T
> length(dice10[(dice10==2) | (dice10==6)]) #
Counts no. Of 2's or 6's
[1] 4
> length(dice10) # Counts total outcomes
[1] 10
Probability of occurrence of 2's and 6's
> p26 = length(dice10[(dice10==2) |
(dice10==6)])/length(dice10)
> p26
[1] 0.4
```

Now, you can recall that you have done two types of logical operators, one was here "and", and other was here "or", or. And they were indicated by the symbol this "&" this vertical line | for "or". So now, if you tried to to execute the computation of this probability on the R software. So, we have this data set dice10, same data set. And now you can see here 6 is occurring only here and 2 is occurring 1, 2, 3 places.

And now, in this case, well, either 2 is occurring 6 is occurring, but definitely 2 and 6 are not occurring together, but definitely means if you try to take an experiment of two dice and then if you roll that and if you try to record the numbers, the minimum number that can be obtained will be 11, and the maximum number that can be obtained is 66. So, if you try to generate the random numbers between 11 and 66 and try to compute the probability that 2 and 6 are coming together, they are occurring together. Then possibly I can also compute the probability of occurring only 2, only 6 or say 2 and 6 together.

But anyway, means my idea here is to give you a simple illustration that how you can compute these probabilities after that, it depends on in your programming capability that how you can do it, but what I gave you the confidence that these things are doable. So, now I try to use here the logical operator here "or" and I try to count here that what are the places where dice10 has the elements say 2 that mean dice10 is logically equal to 2 and we also want to count that what are the places where dice10 has the numbers 6.

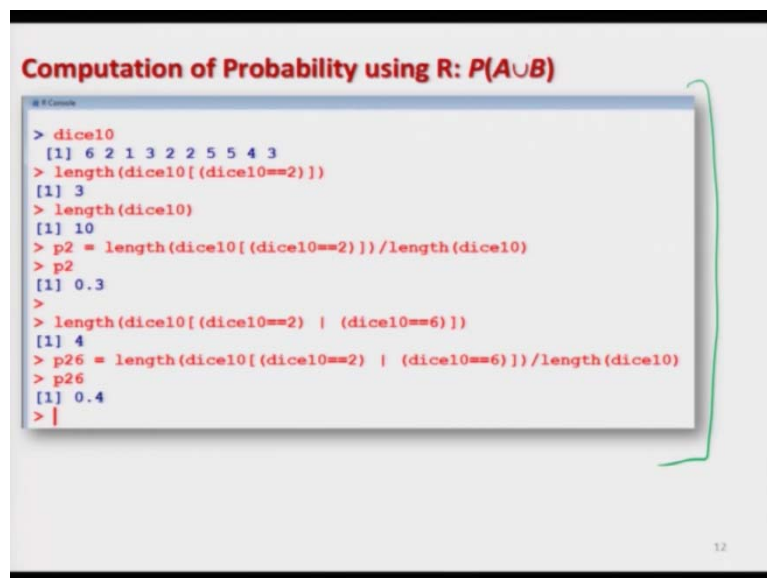
And then we want to operate this or operator on both the numbers. But this is going to give you the answer in terms of TRUE and FALSE. For example, it will now give you the answer

as T then here T then for 1, it will be FALSE; for 3, it will be FALSE; for 2, it will again be TRUE too and so on.

So, now, it will try to count the number of TRUES and they will come out to be here 4, if you try to compute the length of that outcome. Now, you simply have to compute the length of the dice vector which is here 10. And now, if you try to compute the probability of 2 or 6 which I am indicating by here p26 simply try to divide these two probabilities, this probability and this probability right and you will get here 0.4.

And you can see that this value here is 0.4 can be very easily obtained here that if you try to see here there are 1, 2, 3, 4 number which are satisfying this condition and so, this number becomes here 4 divided by total number of values, which is here 10 and this is 0.4. So, you can see here that it is not difficult to compute the probabilities of unions.

(Refer Slide Time: 17:55)



```
Computation of Probability using R:  $P(A \cup B)$   
> dice10  
[1] 6 2 1 3 2 2 5 5 4 3  
> length(dice10[(dice10==2)])  
[1] 3  
> length(dice10)  
[1] 10  
> p2 = length(dice10[(dice10==2)])/length(dice10)  
> p2  
[1] 0.3  
> length(dice10[(dice10==2) | (dice10==6)])  
[1] 4  
> p26 = length(dice10[(dice10==2) | (dice10==6)])/length(dice10)  
> p26  
[1] 0.4  
> |
```

And similarly, as I said, the other probabilities are not difficult. And this is a screenshot of the same operation which I have just shown you.

(Refer Slide Time: 18:03)

Computation of Probability using R: $P(A | B)$

In case of computation of conditional probability,

$$P(A | B) = \frac{P(AB)}{P(B)}$$

How to compute $P(AB)$,
Hypothetically

```
length(d10[d10==A] & [d10==B])
```

We know how to compute all the involved expressions.
Using logical operators, one can also compute them directly also by writing a suitable function.

Height	Age
151cm	35
160cm	35
181cm	33

Now, suppose if you want to compute the conditional probability. Now, computing conditional probability either you can directly use the logical operators that you want to have more than one condition that what are the values of X for which the number of Y values are equal to suppose 2. And how you can compute it? For example, if you have here two random variables X and Y, or say here height and say here age, these are two variables.

Now, you are saying here that height is 151-centimeter, 160-centimeter, 181-centimeter and so on. Suppose the age here is say 35 years, 35 years, 33 years and so on. So, means you want to suppose condition on the age. So, you can simply find out the value of those X's where Y is equal to for example, 35. It is not difficult, just by using the logical operators you can do it.

And then you have to simply count that how many such numbers are there for a corresponding to which the age is 35. Once you can find out that length, then you can very easily compute the conditional probability by using the same result that probability of A intersection B divided by probability of B will compute you the probability of A given B.

Now, how to compute the probability of AB? I have given you an example here and hypothetically if you try to see in this event that we just conducted although that was a discrete experiment and there are no interaction between A and B. So, but in case if you try to say that there is some data vector.

So, hypothetically if you try to see if you have taken this detailed data vector here like this, you simply try to count that how many are there A and how many there are B and then you try to count on the total number of observations by using the command here length. So, that

will give you the absolute frequency and if you try to divide it by the total amount of observation that will give you the conditional probability that A and B are occurring together. And the only thing here is this, that you have to use here the operator "and" instead of "or". That is the vertical line.

So, now we know how to compute all the involved expression probability of A intersection B, probability of B. So, using the logical operators, one can compute them directly also using the suitable function. Or you can use this definition also and can compute the probability of A given B or probability of B given A, whatever you want. Well, I am not doing it here, because it just depends on your capability. My job is only to give you an idea that how are you going to do it. But definitely, if you practice more, this is definitely going to help you more.

(Refer Slide Time: 21:02)

```

> dice10 = sample(c(1,2,3,4,5,6), size=10, replace = T)
> dice10
[1] 3 3 4 5 4 3 5 1 2 2
> length(dice10[(dice10==2)])
[1] 2
> dice10==2
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
> dice10[(dice10==2)]
[1] 2 2
> length(dice10)
[1] 10
> length(dice10[(dice10==2)])/length(dice10)
[1] 0.2
> dice1000 = sample(c(1,2,3,4,5,6), size=1000, replace = T)
> length(dice1000[(dice1000==2)])/length(dice1000)
[1] 0.2
> length(dice1000[(dice1000==2)])/length(dice1000)
[1] 0.169
>

```

Computation of Probability using R: P(A)

```

> dice10
[1] 6 2 1 3 2 2 5 5 4 3

```

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
T, F T, F T, F T, F T, F T, F T, F T, F T, F

logical equality

```

> length(dice10[(dice10==2)]) # Counts no. Of 2's
[1] 3

```

↑ ↑ ↑
T, F T, F T, F

```

> length(dice10) # Counts total occurrence
[1] 10

```

Probability of occurrence of 2's

```

> p2 = length(dice10[(dice10==2)])/length(dice10)
> p2
[1] 0.3

```

→ $\frac{3}{10}$

Now, look, let me try to come to the R console and try to show you here the same example that how these things are working. So, let me try to generate here the 10 observations out of 1, 2, 3, 4, 5, 6 and you can see here, these are the values which have here come. So, there are only two times the 2 are coming. And if you try to see here, I try to now compute this expression.

So, first I try to compute this expression and then I will try to show you what exactly it is doing. So, if you try to see the final values coming out to be 2, but how does it take coming out to be 2? Let me try to first show you the outcome of case expression. You can see here this is giving you FALSE, FALSE, FALSE, FALSE and last two values are here TRUE, because there is only here 2 at only two places.

And then if you try to compute here this command which is trying to find out what are the values which are TRUE. So, you can see here this is giving you these two values, 2 and 2, they are reported here. And then if you try to find out the length of this vector, this is obviously here 2.

So, now, in case if you try to find out here the this here, length of a dice10, you know that this is going to be here simply 10. And then if you try to compute here, this probability here, it is just going to be the division of both the values and it is coming out to be here 0.2. No issues. And you can see here 2 out of 10, these are the values.

And similarly, if you try to repeat the experiment for say a large number of time, suppose I try to repeat this experiment for say here 1000 time. Then you just cannot count these things manually. And this data comes over here. And now if you try to compute this probability here directly you can see here this is again coming out to be here 0.2.

But this is not the probability of dice1000 observation. This is only for the dice10. And now, if you try to compare it, I am trying to make it here for 1000 observation and you see how it is changing. You can see here, now, this is coming out to be 0.169, which is more close to 1 upon 6.

(Refer Slide Time: 23:18)

```
File Edit View Misc Packages Windows Help
R Console
> dice10
[1] 3 3 4 5 4 3 5 1 2 2
> length(dice10[(dice10==2) | (dice10==4)])
[1] 4
> length(dice10[(dice10==2) | (dice10==4)]/length(dice10))
[1] 0.4
> length(dice1000[(dice1000==2)])/length(dice1000)
```

```
File Edit View Misc Packages Windows Help
R Console
> dice10
[1] 3 3 4 5 4 3 5 1 2 2
> length(dice10[(dice10==2) | (dice10==4)])
[1] 4
> length(dice10[(dice10==2) | (dice10==4)]/length(dice10))
[1] 0.4
> dice1000
```

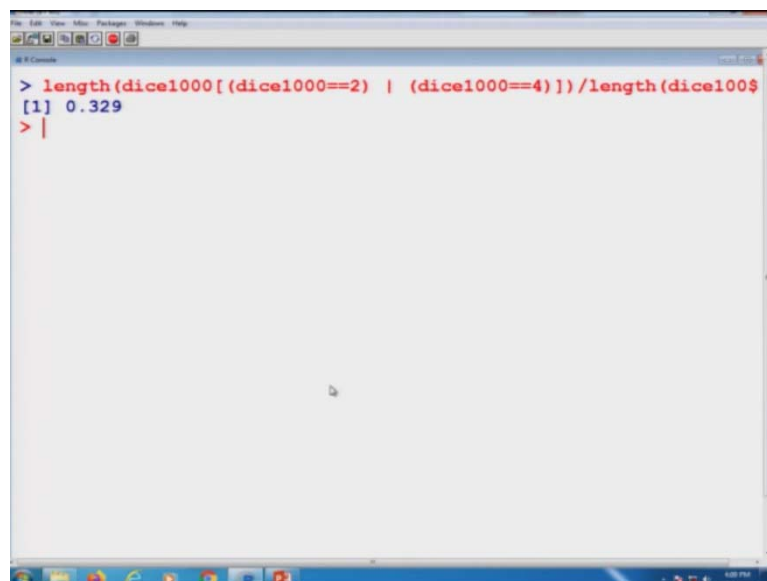
```
File Edit View Misc Packages Windows Help
R Console
[320] 2 1 6 5 5 2 6 2 5 6 4 6 3 1 1 6 3 5 3 6 2 2 3 4 5 6 1 6 2
[349] 5 4 3 4 6 3 3 3 6 2 3 3 2 1 3 6 6 2 2 6 5 3 2 6 3 3 1 5 5
[378] 1 1 5 5 1 2 6 5 1 4 2 3 4 2 4 4 5 3 5 2 3 5 6 4 6 4 4 1 3
[407] 2 5 3 5 2 5 4 4 2 6 1 2 1 1 6 3 1 5 4 1 1 4 5 5 4 2 5 2 1
[436] 4 1 4 3 3 3 2 3 4 3 1 3 5 5 5 4 1 2 5 3 1 3 2 1 2 4 4 3 6
[465] 2 1 6 6 2 1 1 3 5 4 3 5 4 6 4 5 1 6 1 5 1 5 4 3 2 5 4 1 1
[494] 4 1 2 3 1 6 4 5 1 4 5 3 3 1 4 2 3 3 6 5 1 4 1 1 3 2 2 1 1
[523] 6 6 6 1 6 6 5 5 1 6 6 4 3 3 1 3 1 3 4 1 3 2 2 5 2 3 1 3 2
[552] 6 3 6 5 4 6 2 4 1 4 5 3 1 3 2 2 3 3 3 3 3 6 4 2 3 3 3 1 6
[581] 6 2 3 2 5 4 4 4 1 4 5 5 2 2 5 2 3 3 2 1 2 2 5 2 3 5 3 2 2
[610] 3 4 2 5 2 2 6 4 3 3 3 1 5 6 6 5 6 3 3 3 3 3 6 5 6 4 5 6 2
[639] 5 1 6 5 2 4 1 1 6 1 6 6 5 4 6 1 3 1 6 1 2 4 2 5 3 4 4 1 2
[668] 1 3 4 3 6 5 4 1 2 3 3 5 3 2 1 4 2 3 3 4 4 4 1 3 3 3 3 4 1
[697] 5 1 2 3 5 5 3 2 2 5 3 3 3 4 1 4 1 5 6 1 4 6 2 5 2 2 4 5 2
[726] 6 3 3 3 3 5 2 3 4 6 5 2 1 1 5 6 4 5 4 3 6 6 5 4 2 6 4 2 2
[755] 2 5 6 2 1 2 1 5 3 1 5 5 2 4 5 1 3 6 5 6 1 5 6 4 3 2 2 4 6
[784] 6 2 5 1 1 5 2 2 2 4 2 3 1 6 1 1 1 6 4 4 2 3 2 4 1 3 2 1 4
[813] 3 3 4 5 6 2 6 4 4 3 6 1 4 6 4 3 2 2 5 3 1 3 1 5 3 4 2 6 3
[842] 3 2 4 3 1 4 3 4 2 6 6 6 1 4 5 4 5 6 4 6 5 5 1 5 6 1 6 1 2
[871] 5 6 1 5 6 1 5 6 1 5 6 3 1 6 4 3 6 3 5 5 4 6 1 5 3 6 1 5 6
[900] 3 4 2 2 3 6 3 1 4 2 6 3 5 6 5 1 6 4 4 4 3 1 1 1 2 5 3 4 1
[929] 5 1 1 4 4 3 2 3 6 5 4 6 1 1 5 1 5 1 3 4 6 1 2 5 6 6 3 6 6
[958] 1 5 3 3 1 3 4 5 2 4 3 6 1 1 1 1 2 4 3 1 4 5 1 1 4 4 3 1 4
```

So, now, let me try to clear the screen and the data what we have is here dice10 like this and now let me try to do it for two numbers. In the example I have taken it to be 2 and 4, but unfortunately, you will see here that there are no 6. In the example I have taken it for 2 and 6. So, I am trying to take it here for 2 and 4 here.

So, you can see here there are two numbers which are 2 and there are two numbers which are here 4. And if you try to find out the total length, it will come out to be here 4. And if you try to divide this length by length of data vector dice10, you can see here this will come out to be 0.4.

Now, I tried to just repeat this experiment for 1000 time and you can see here this, you had this data vector here, dice1000 that you have done means earlier. So, I try to use it here. And then if you try to have a look on the dice1000 data, it will look like this. That yeah, that is a very big data set.

(Refer Slide Time: 24:23)



```
> length(dice1000[(dice1000==2) | (dice1000==4)])/length(dice1000$)
[1] 0.329
> |
```

But then anyway, my concern is this I want to compute this probability. So, I try to change here the data set to be here dice1000. And if you try to see here now, from this data set, it is trying to count that how many 2's and 4's are there and this probability is coming out to be 0.329. So, because this is equal to the probability of 2 plus probability of 4 which are equal to nearly equal to 0.16 plus 0.16. So, it is coming it will be very close to 0.32.

So now we come to an end to this lecture and you can see this I have tried my best to give you a some logic, some idea that how you can compute these probabilities on the basis of

given set of data, given set of process. Now, it is up to you that how much you can think about it, how much you can create a logic and how efficiently you can compute the different types of probabilities by thinking in terms of logical operators or something else.

So, I would say why don't you take some very simple examples and try to see how you can execute the simple probabilities like probability of A, probability of A union B, probability of A intersection B etc. Just take very simple example, try to however, you can take a very small data set that is artificially created data set, where you know that what is happening and try to see the way you are trying to compute it are, they matching with the result which you are obtaining from your theory.

Once these two results matches you will get more confidence when you are trying to deal with much bigger data sets. Whatever is the value as an outcome is coming, you will have more faith on it. So, you try to practice it and I will see you in the next lecture till then, goodbye.