Essentials of Data Science with R Software - 2
Sampling Theory and Linear Regression Analysis

Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology Kanpur

Sampling Theory with R Software
Lecture - 27
Sampling for Proportions and Percentages
Sampling for Proportions with R

Hello friends, welcome to the course Essentials of Data Science with R Software 2 where we are trying to learn the topics of Sampling Theory and Linear Regression Analysis. In this module, we are going to continue with the Sampling Theory with R Software and we will continue with the chapter where we are trying to learn the Sampling Theory for Proportions and Percentages now with R software.

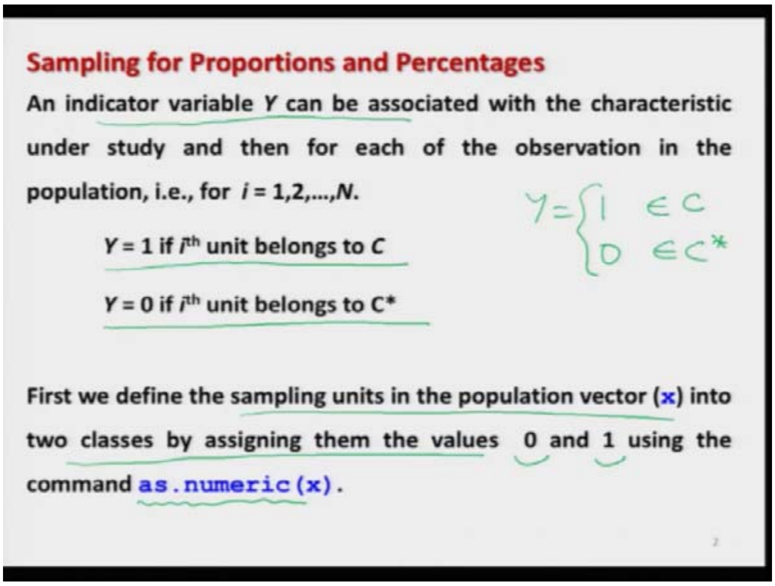
So, in the last two lectures, we consider the simple random sampling for qualitative variables and we had defined the sample proportion as an estimator for population proportion. We found the standard errors of a sample proportion and in this lecture, I am going to show you that how you can implement the estimation of population proportion as well as finding out the standard errors when the samples have been drawn by simple random sampling with replacement and without replacement.

So, in order to show you all these things on the R software, first I would try to create a population. Once you create a population yourself, then you know what is happening in the population and then, you will implement all the tools whatever we have developed in the last lecture on the R software. And when you are trying to implement the computation of proportions in the R software, you have to remember first one thing that is very important in the case of proportion and percentages.

Whatever observations you are getting, they are going to be in terms of yes, no, agree, disagree etc. So, they will be converted into 0 and 1's. Once you convert into 0 and 1's, then we need to convert that data into a numeric vector. Why? Because I will show you that the population will be converted at some intermediate steps in the terminologies of true and false. So, those true and false will be converted into the real population.

But in practice, this may not happen. In practice, you will be getting a data in some good, bad, yes, no so, you have to just decide once that what are your indicators whether good is represented by 1 or 0, bad is represented by 1 or 0. So, this is what you have to keep in mind. So, in the first part of the lecture, I will try to create a population and then, I will try to implement all the expressions that we have developed in the last lecture. So, let us begin our lecture from this slide.

(Refer Slide Time: 03:34)



Sampling for Proportions and Percentages

An indicator variable Y can be associated with the characteristic under study and then for each of the observation in the population, i.e., for $i = 1,2,\ldots,N$.

$Y = 1$ if i^{th} unit belongs to C

$Y = 0$ if i^{th} unit belongs to C*

$$Y = \begin{cases} 1 & \in C \\ 0 & \in C^* \end{cases}$$

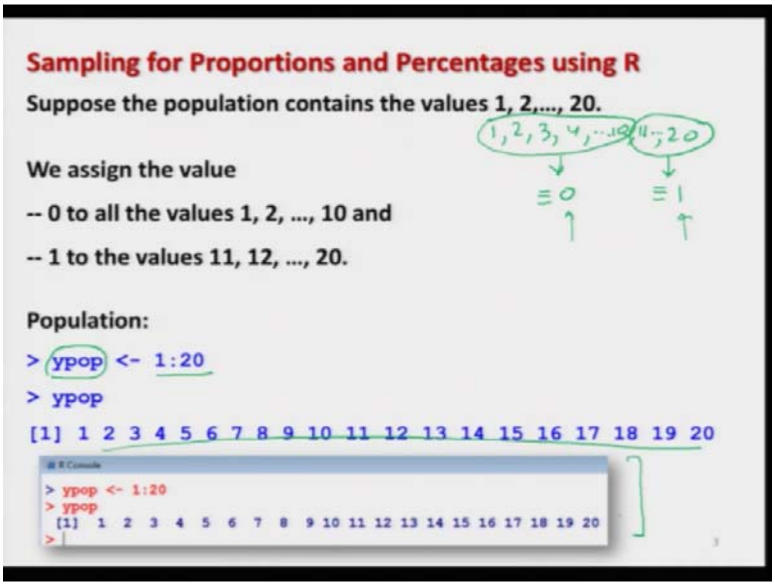
First we define the sampling units in the population vector (**x**) into two classes by assigning them the values 0 and 1 using the command `as.numeric(x)`.

So, we already have learned that whenever we have a qualitative variable, the observations on the qualitative variable are converted in a quantitative variable which we had denoted by Y and Y was defined in such a way such that Y takes value 1 and 0 and 1 will indicate the units which belong to the class C and 0 will indicate the unit which belongs to the complementary of C class which is indicated by C*.

So, now for each of the unit in the population Y_1, Y_2,…,Y_N now, I can define here Y equal to 1 if the ith unit belongs to C and Y equal to 0 if ith unit belongs to C*, ok.

So, now, first we try to create the population. So, in order to define the sampling units in the population vector which we are going to denote by x which is further going to be classified into two classes. So, the values will be indicated by 0 and 1 and this can be obtained using the command as dot numeric and inside the parenthesis you have to give the population vector or the data vector.

(Refer Slide Time: 04:59)



Sampling for Proportions and Percentages using R

Suppose the population contains the values 1, 2,..., 20.

We assign the value

-- 0 to all the values 1, 2, ..., 10 and

-- 1 to the values 11, 12, ..., 20.

Population:

```
> ypop <- 1:20
> ypop
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

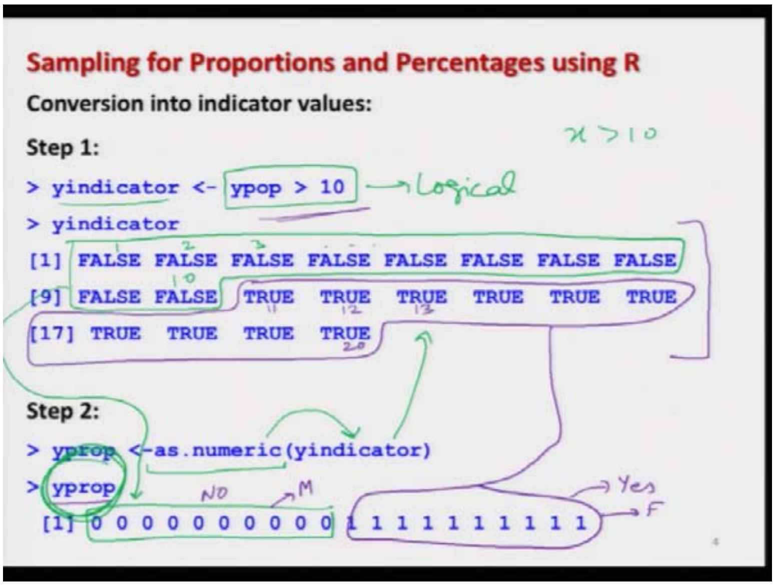
```
> ypop <- 1:20
> ypop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
>
```

So, let me try to take an example first. Suppose I consider a population which is simply consisting of the numbers 1, 2, 3, 4 up to 20 right and somewhere here, it will be 10 and then 11 and then 20.

So, what I try to do here that I try to divide these numbers 1 to 10 say into one class and we denote it by 0 and all the values 11, 12, 13, 14 up to 20 they are assigned to another class, complementary class and they will be denoted by here 1. You can take reverse also means you can denote the first group by 1 and second group by 0 also that is your choice ok.

So, here data vector ypop means population of y as 1 to 20 and you can see here these are my population values and this is the screenshot here. So, that is simple straight forward.

(Refer Slide Time: 06:08)



Sampling for Proportions and Percentages using R

Conversion into indicator values:

Step 1:

```
> yindicator <- ypop > 10
> yindicator
[1]  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
[9]  FALSE  FALSE  TRUE   TRUE   TRUE   TRUE   TRUE   TRUE
[17] TRUE   TRUE   TRUE   TRUE
```

Step 2:

```
> yprop <-as.numeric(yindicator)
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
```

Now, I need to convert this data into two classes. So, I try to find out the population values which are greater than 10 and I try to assign all such values or the outcome of this command ypop is greater than 10 and a new vector yindicator.

You can recall that when we did the some basic fundamental of the R software, then we had learnt that what is the meaning of some data vector x greater than 10 or x greater than 20 so, that was a logical operator. So, I am trying to use here a logical operation.

So, what will happen here? That all the values in the y population which is ypop, the numerical values from 1 to 20 out of those thing, first 10 values that means, number 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 they will be denoted as FALSE. So, this first value is 1, second value is 2, third FALSE is 3 and so on and the last FALSE here is 10 and the remaining values which are going from 11 to 20, they will be denoted by here TRUE because what it is trying to say?

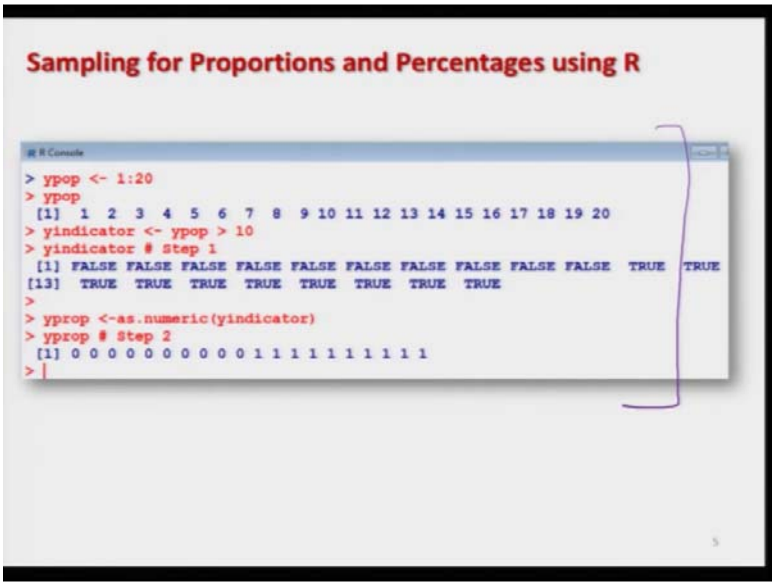
When I say ypop is greater than 10 that means, all the values which are greater than 10, this condition is TRUE so, this is what is indicated by here these TRUE. So, this TRUE will be indicated the value 11, second value will be 12, third value will be 13 and the last value will be here 20, but this outcome what is obtained from the yindicator this is it this is in terms of TRUE and FALSE. So, I need to convert it into a numerical vector.

So, in order to convert it into a numerical vector, we will use here a command as numeric, as dot numeric. So, the command as dot numeric will change FALSE into 0 and TRUE into 1.

So, you can see here once I try to operate this command on this data vector yindicator that we have obtained here, all the values which are here FALSE, they will become here 0, right and whatever are the values which I had obtained under TRUE, they are converted here into 1's and I try to denote these values as yprop.

So, yprop is my resulting population. The population in which first 10 values are 0 and say remaining 10 values are 1. So, that may indicate that these are the people who said no to the proposal and these are the people who said yes to the proposal or this is a group of students which are male, this is the group of student which are female and so on. So, now, my population is with us now.

(Refer Slide Time: 09:39)

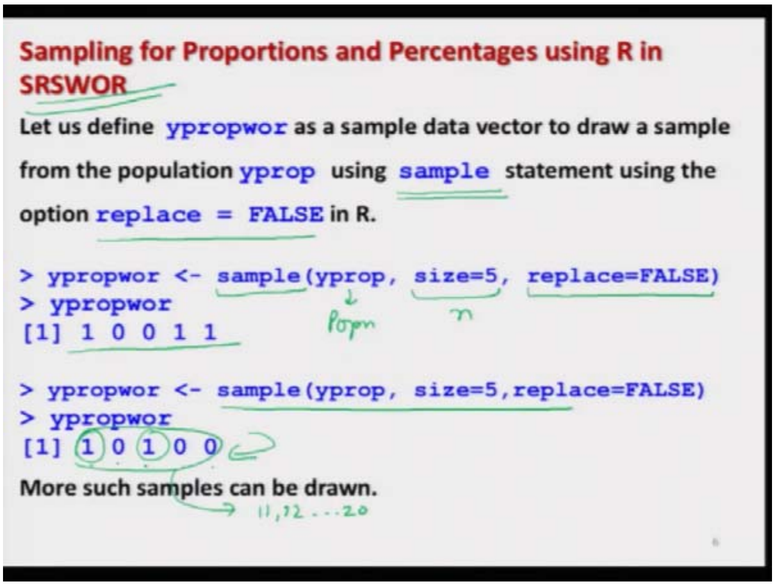


Sampling for Proportions and Percentages using R

```
> ypop <- 1:20
> ypop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> yindicator <- ypop > 10
> yindicator # Step 1
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
>
> yprop <-as.numeric(yindicator)
> yprop # Step 2
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
> |
```

And we would like to draw here a sample, and this is here that you can see this is the screenshot. So, if you try to do it yourself on the R console, you will get this one although I will try to do it at a later stage.

(Refer Slide Time: 09:51)



Sampling for Proportions and Percentages using R in SRSWOR

Let us define `ypropwor` as a sample data vector to draw a sample from the population `yprop` using `sample` statement using the option `replace = FALSE` in R.

```
> ypropwor <- sample(yprop, size=5, replace=FALSE)
> ypropwor
[1] 1 0 0 1 1
```

Popn

n

```
> ypropwor <- sample(yprop, size=5,replace=FALSE)
> ypropwor
[1] 1 0 1 0 0
```

More such samples can be drawn.

→ 11,12 ...20

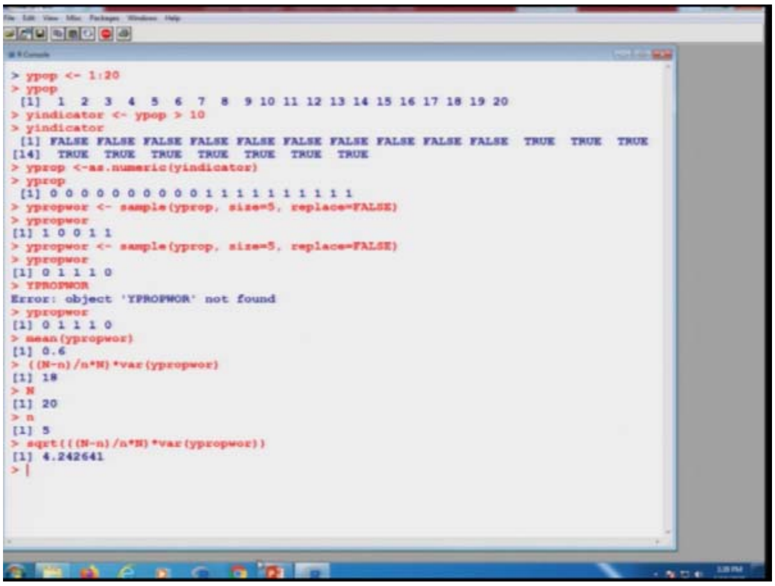
Now, what I have explained you in the last lecture that we will use simply simple random sampling to draw a sample from this population. So, my population here is yprop consisting of ten 0s and ten 1s.

So, from there, I would try to obtain a sample. So, in order to obtain the sample, I will be using the same command which I use in the case of simple random sampling, which is sample, sample and then, there will be an option that replace is equal to TRUE or FALSE that will indicate whether we want the sample with replacement or without replacement.

So, at this stage, first we try to consider the simple random sampling without replacement, right. So, I try to use the same command that we discussed earlier sample, then the population from where we would like to sample, sample size here n so, I am trying to choose here sample of size n that means, from the population yprop. And since I want SRSWOR so, I am using here replace is equal to FALSE; that means, please do not replace the value.

Now, once I try to execute this command on the R console, here I am got; here I am getting this sample 1, 0, 0, 1, 1 and similarly, if you try to repeat this sample here, you will get here another sample 1, 0, 1, 0, 0. So, definitely this is a random sample. So, means every time you try to repeat this command, you will get a different sample, right. So, first let me try to show you this thing on the R console, right ok.

(Refer Slide Time: 12:05)



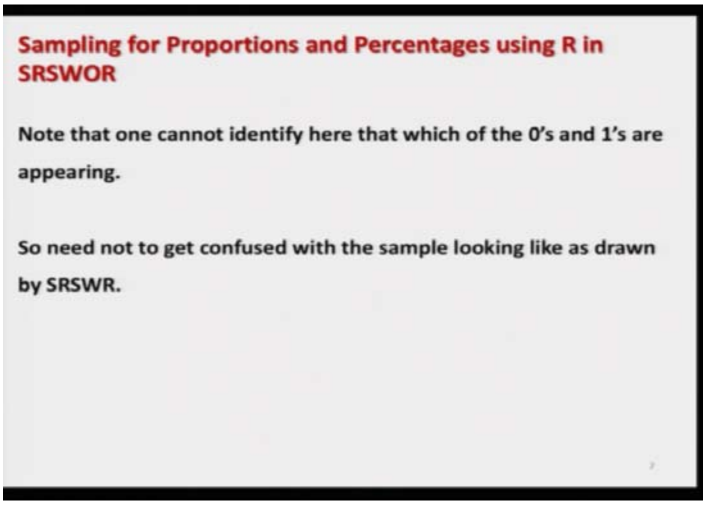
```
> ypop <- 1:20
> ypop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> yindicator <- ypop > 10
> yindicator
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
[14]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> yprop <-as.numeric(yindicator)
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
> ypropwor <- sample(yprop, size=5, replace=FALSE)
> ypropwor
[1] 1 0 0 1 1
> ypropwor <- sample(yprop, size=5, replace=FALSE)
> ypropwor
[1] 0 1 1 1 0
> YPROPWOR
Error: object 'YPROPWOR' not found
> ypropwor
[1] 0 1 1 1 0
> mean(ypropwor)
[1] 0.6
> ((N-n)/n*N)*var(ypropwor)
[1] 18
> N
[1] 20
> n
[1] 5
> sqrt(((N-n)/n*N)*var(ypropwor))
[1] 4.242641
> |
```

So, you can see here that first I try to define here the population R console so, you can see here this is your population the number from 1 to 20. Next command in this I am trying to use here that try to choose all those values from this population which are greater than 10. So, I try to do it here and you see here I get yindicator as like this, this is the same command which you have obtained here.

And now, I try to convert this thing into numerical vector. So, I try to convert these values into numerical vector. So, you can see here yprop, ok. Now, when I try to draw the sample from here so, I try to use the command sample and I try to draw a sample of size 5. So, you can see here this is the sample yproportion without replacement, right.

And if you try to repeat this command, you will get here a different sample. So, that is the reason that whatever results I have presented here, when you try to execute it on the R console, they may not exactly match because every time, I draw the sample that is going to be different and, in this slide, whatever results I have presented, they are based on the sample which I drawn when I was trying to prepare the slides.

(Refer Slide Time: 13:44)



Sampling for Proportions and Percentages using R in SRSWOR

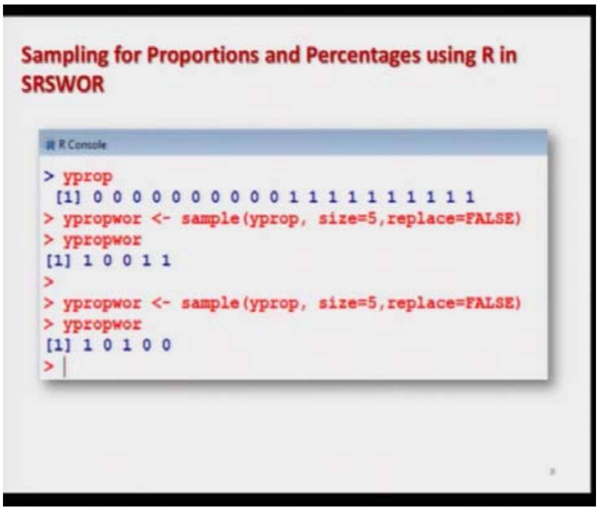
Note that one cannot identify here that which of the 0's and 1's are appearing.

So need not to get confused with the sample looking like as drawn by SRSWR.

And one thing as I said earlier, you can now notice here that in the sample also if you just try to look at the sample and if you do not consider the; and if you do not consider the population well, you do not know, then by looking at this 1, 0, 1, 0, 0, you cannot identify whether these 1’s, 1, 1, they are corresponding to which of the values out of 11, 12 up to 20.

You cannot identify it by looking at the value that this 1 is corresponding to which of the value for that you have to do something more, right. If you remember, we had done a similar example where we took the height of the students and we did everything. So, you need to do a little bit more computation in the R using your knowledge of R right.

(Refer Slide Time: 14:45)

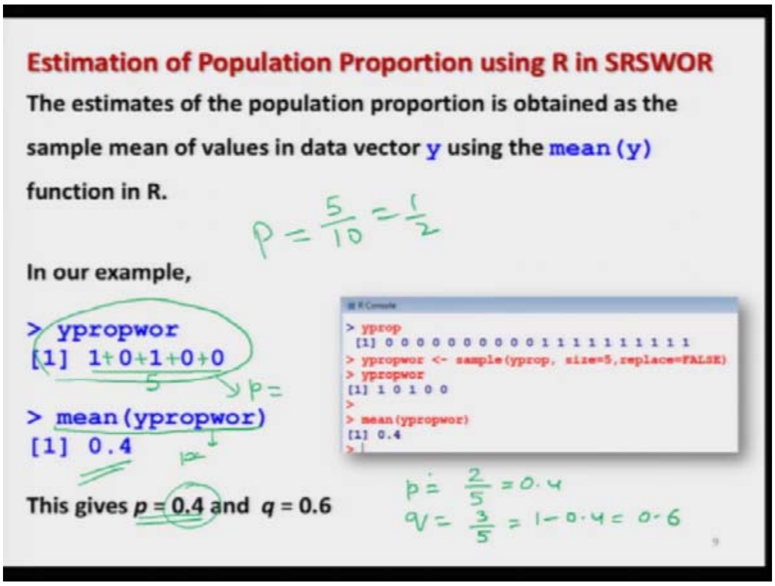


Sampling for Proportions and Percentages using R in SRSWOR

```
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
> ypropwor <- sample(yprop, size=5,replace=FALSE)
> ypropwor
[1] 1 0 0 1 1
>
> ypropwor <- sample(yprop, size=5,replace=FALSE)
> ypropwor
[1] 1 0 1 0 0
> |
```

So, please do not get confused now. So, this is my screenshot and now.

(Refer Slide Time: 14:47)



Estimation of Population Proportion using R in SRSWOR

The estimates of the population proportion is obtained as the sample mean of values in data vector **y** using the **mean(y)** function in R.

$P = \frac{5}{10} = \frac{1}{2}$

In our example,

```
> ypropwor
[1] 1+0+1+0+0
> mean(ypropwor)
[1] 0.4
```

$\frac{1+0+1+0+0}{5} \rightarrow p =$

This gives $p = 0.4$ and $q = 0.6$

```
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
> ypropwor <- sample(yprop, size=5,replace=FALSE)
> ypropwor
[1] 1 0 1 0 0
>
> mean(ypropwor)
[1] 0.4
>
```

$p = \frac{2}{5} = 0.4$

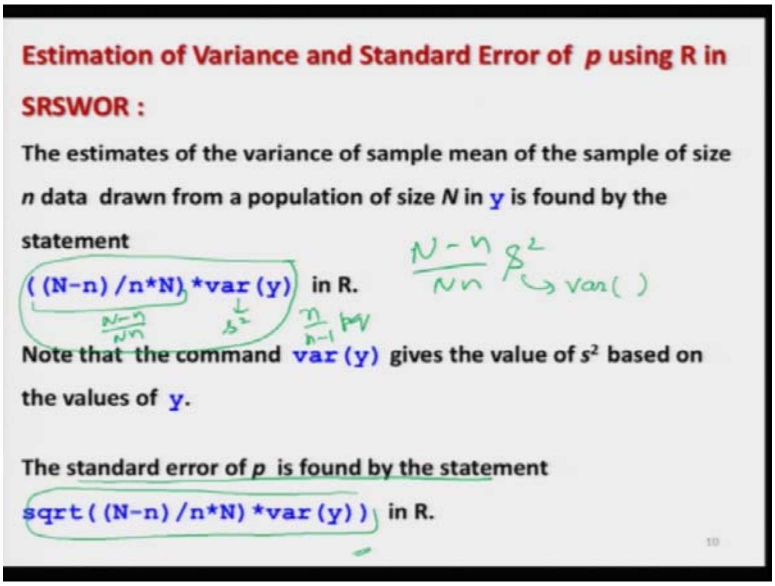
$q = \frac{3}{5} = 1 - 0.4 = 0.6$

So, now, I fix my sample here that my sample here is like this 1, 0, 1, 0, 0. So obviously, if you want to estimate the population proportion here, population proportion well in practice that will be unknown to you, but in this case, this is 5/10 which is 1/2 and now from the sample, you would like to estimate the value ofP. So, this I am estimating here is a p that is the number of 1's and 0's. So, you can see here one, the number of 1's here are 2 divided by here 5.

So, this is equal to here 0.4 and this is the value of p, right. And what is here q? q either you can obtain by number of 0's which is here 3 divided by 5 or this is also 1 - 0.4 which is here 0.6. So, you can see here when you are trying to compute here the mean of this sample data, this is trying to give you some 1 plus 0 plus 1 plus 0 plus 0 divided by 5. So, this comes out to be 2/5 which is 0.4.

So, you can see here that the population estimate here is 0.4 and you can see since you are using a good estimator so, this value is very close to 0.5 a value and even if you try to take some more samples, I will show you when I come to the R console that this value will not change much because you are using a; because you are using an estimator with good statistical properties.

(Refer Slide Time: 16:46)



Estimation of Variance and Standard Error of *p* using R in SRSWOR :

The estimates of the variance of sample mean of the sample of size *n* data drawn from a population of size *N* in `y` is found by the statement

`((N-n)/n*N)*var(y)` in R.

$\frac{N-n}{Nn}$ s^2 $\frac{n}{n-1}pq$

$\frac{N-n}{Nn}s^2$ ↳ var( )

Note that the command `var(y)` gives the value of s^2 based on the values of `y`.

The standard error of *p* is found by the statement

`sqrt((N-n)/n*N)*var(y))` in R.

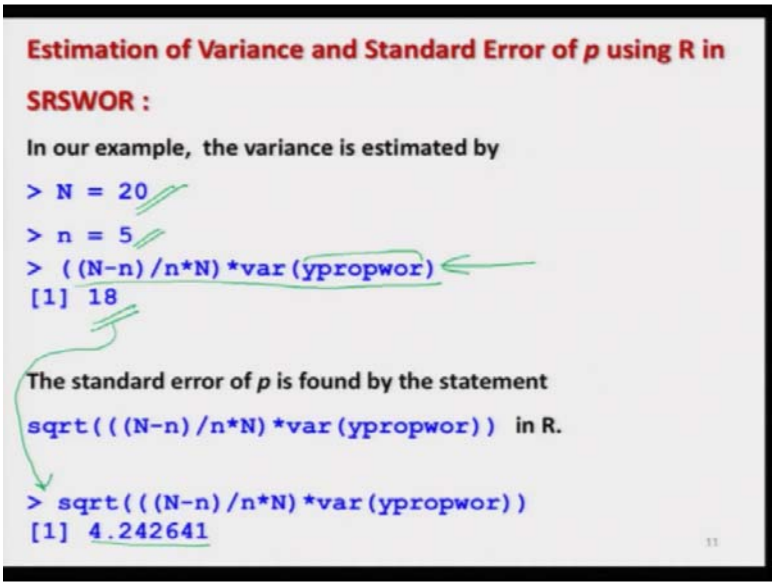
10

Now, we already had obtained the simple random the variance of $\bar{y}$ under simple random sampling under WOR. So, this expression was $\frac{N-n}{Nn}S^2$ and estimate was as^2 and we know that this s^2 is being computed in the R by the command variance.

So, what I can do? I can write here this quantity which is here (N – n)/Nn and this will give you as^2 and which is the same quantity say $\frac{n}{n-1}$pq as we discussed in the earlier lecture right.

So, and if you want to find out the standard error so, you simply have to take the positive square root of this quantity that is the variance of p. So, here I am trying to write down the square root of the variance of here p which is here like this.

(Refer Slide Time: 18:00)



Estimation of Variance and Standard Error of *p* using R in SRSWOR :

In our example, the variance is estimated by

```
> N = 20
> n = 5
> ((N-n)/n*N)*var(ypropwor)
[1] 18
```

The standard error of *p* is found by the statement

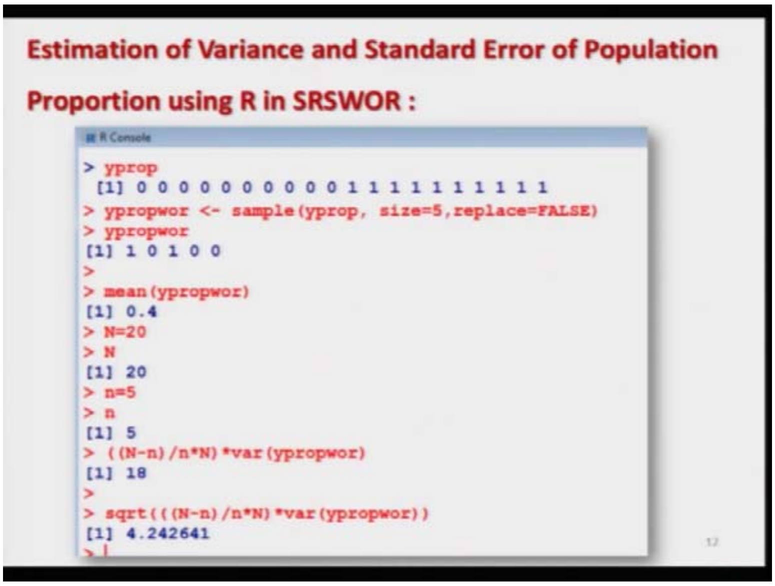
`sqrt(((N-n)/n*N)*var(ypropwor))` in R.

```
> sqrt(((N-n)/n*N)*var(ypropwor))
[1] 4.242641
```

So, now, you have seen that from the given sample, we can compute the variance of sample proportion as well as the standard error. So, since we have taken a population of size here 20 so,N is here 20, an is sample size 5 so, I can compute this quantity, this variance of here p and the data what we have considered it is stored in the data vector ypropwor. And if I want to find out the standard error, I simply have to take the positive square root of this value of variance.

So, if you try to execute this command on the R console, we get here the variance of p as 18 under SRSWOR and if you want to find out the standard error, so, you simply have to take the square root of this expression here and you can see that this quantity comes out to be 4.24 right. So, this is how you can see finding out the estimate of variance or standard error is not difficult at all.

(Refer Slide Time: 19:06)

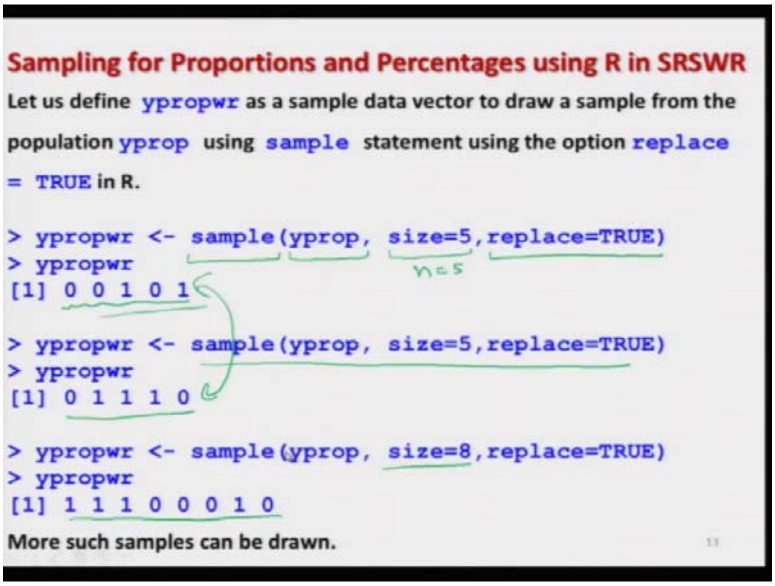


Estimation of Variance and Standard Error of Population Proportion using R in SRSWOR :

```
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
> ypropwor <- sample(yprop, size=5,replace=FALSE)
> ypropwor
[1] 1 0 1 0 0
>
> mean(ypropwor)
[1] 0.4
> N=20
> N
[1] 20
> n=5
> n
[1] 5
> ((N-n)/n*N)*var(ypropwor)
[1] 18
>
> sqrt(((N-n)/n*N)*var(ypropwor))
[1] 4.242641
>
```

And here, this is the screenshot whatever I have presented here.

(Refer Slide Time: 19:14)



Sampling for Proportions and Percentages using R in SRSWR

Let us define ypropwr as a sample data vector to draw a sample from the population yprop using sample statement using the option replace = TRUE in R.

```
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 0 1 0 1

> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 1 1 1 0

> ypropwr <- sample(yprop, size=8,replace=TRUE)
> ypropwr
[1] 1 1 1 0 0 0 1 0
```

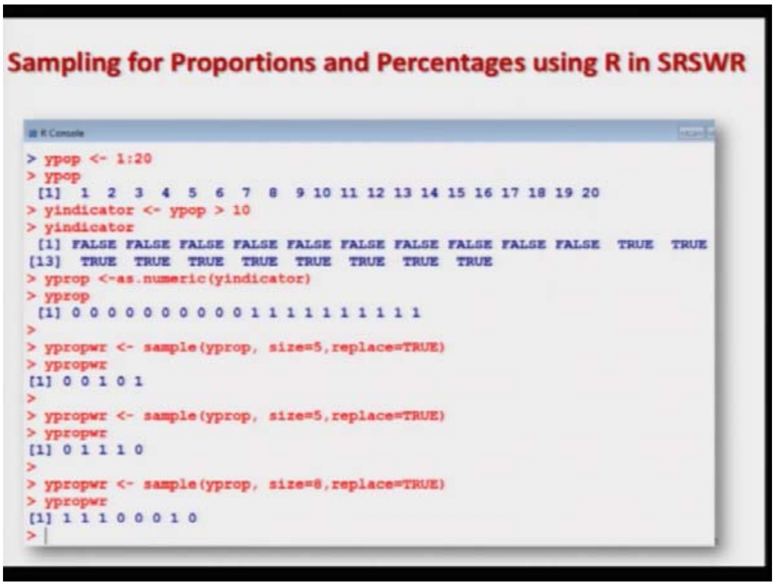
More such samples can be drawn.

So, now, I come to the second case SRSWR. So, I try to consider the same population and I try to draw here a sample, but WR. So, I will use the same command sample and the population is here the same yprop, sample size is the same as n is equal to 5 and since we are going for SRSWR so, I will re use here the option replace is equal to TRUE and when I try to execute this command on the R console, I have here a sample 0, 0, 1, 0, 1.

And now, I will try to explain you here and I would like to draw your attention by looking at this sample, you cannot identify also that whether this sample is SRSWR or WOR because there are only two values 0 and 1 right and also you do not know that this 0 and 1 corresponding to which of the unit population unless and until you try to do it in a more systematic way using the command from data frames ok.

Similarly, if you try to repeat this command and you will get here another sample which is different from the first sample and similarly, if you try to increase the sample size also, sample size is now here 8, then executing the same command, we get here a sample of size 8 and such more samples can be drawn. So, I will now try to fix my sample and I will try to do some computation.

(Refer Slide Time: 20:53)

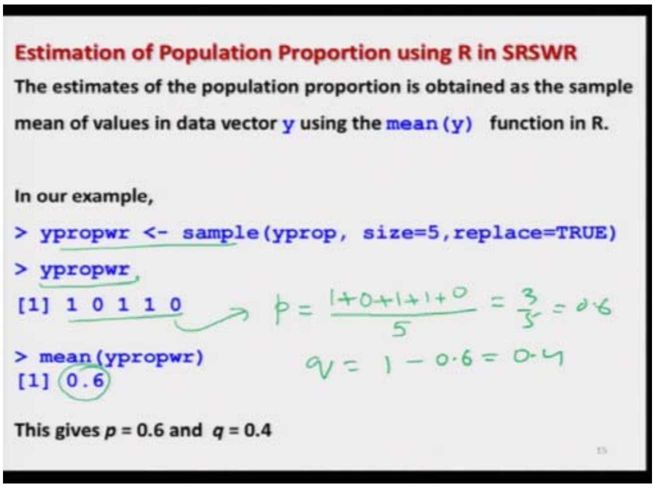


Sampling for Proportions and Percentages using R in SRSWR

```
> ypop <- 1:20
> ypop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> yindicator <- ypop > 10
> yindicator
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> yprop <-as.numeric(yindicator)
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
>
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 0 1 0 1
>
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 1 1 1 0
>
> ypropwr <- sample(yprop, size=8,replace=TRUE)
> ypropwr
[1] 1 1 1 0 0 0 1 0
> |
```

And, but when I go to the R console to demonstrate all this computation, my result will be varying from the result which I have represented in my slides because as said earlier, these results are the were obtained from the sample which was drawn when I was trying to create these slides. So, this is here the screenshot.

(Refer Slide Time: 21:19)



Estimation of Population Proportion using R in SRSWR

The estimates of the population proportion is obtained as the sample mean of values in data vector y using the mean(y) function in R.

In our example,

```
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 1 0 1 1 0
> mean(ypropwr)
[1] 0.6
```

$$p = \frac{1+0+1+1+0}{5} = \frac{3}{5} = 0.6$$

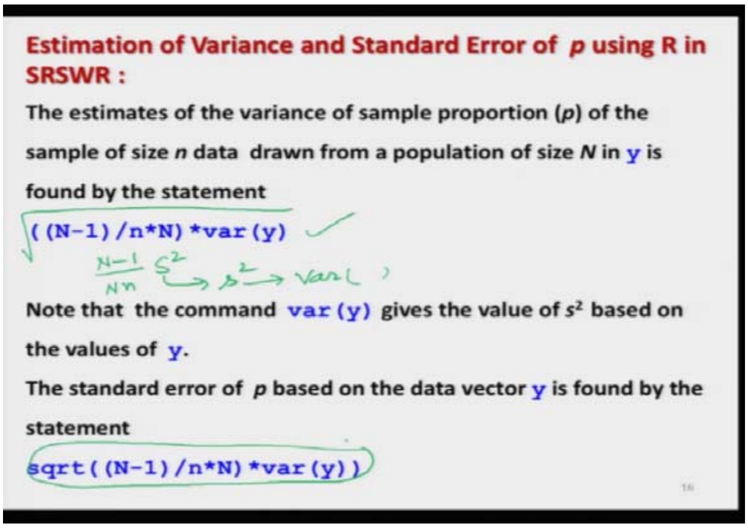
$$q = 1 - 0.6 = 0.4$$

This gives $p = 0.6$ and $q = 0.4$

So, now, I draw a sample by SRSWR and I fix it here as say 1, 0, 1, 1, 0 so, this is my ypropwr means y proportion with replacement and now from here, we know that in order to estimate the population proportion, the sample proportion is an unbiased estimator of the population proportion, so we use it.

So, sample proportion is simply the sample mean of the values in the sample which is 1 + 0 + 1 + 1 +0 divided by 5 and this comes out to be here 3/5 which is 0.6 and q will become obviously, here 1 - 0.6 is equal to 0.4. So, you can see here that the estimate of population proportion comes out to be 0.6 and it is not difficult to find, you simply have to operate the mean command.

(Refer Slide Time: 22:16)



Estimation of Variance and Standard Error of p using R in SRSWR :

The estimates of the variance of sample proportion (p) of the sample of size n data drawn from a population of size N in y is found by the statement

```
((N-1)/n*N)*var(y)
```

$$\sqrt{} \quad \frac{N-1}{Nn} s^2 \rightarrow s^2 \rightarrow var(\ )$$

Note that the command var(y) gives the value of s^2 based on the values of y.

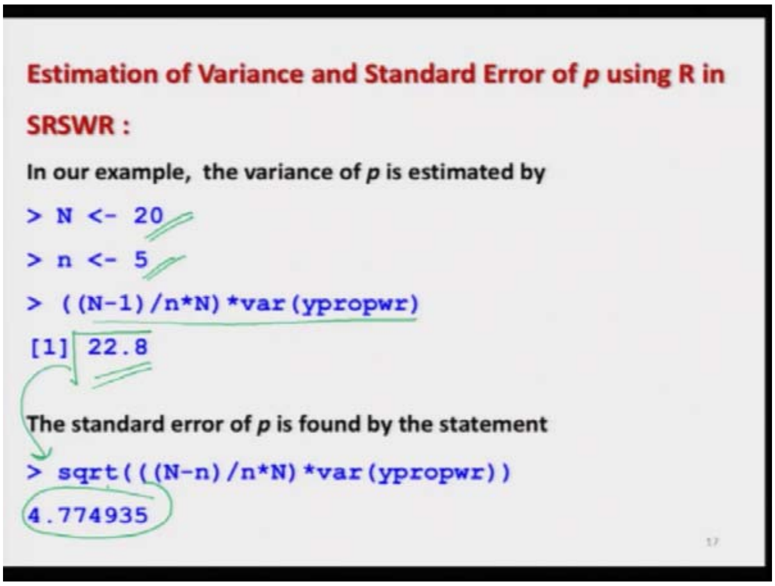
The standard error of p based on the data vector y is found by the statement

```
sqrt((N-1)/n*N)*var(y))
```

And similarly, now we try to find out the estimate of variance. So, if so, you may recall that under SRSWR the variance of sample mean is $\frac{N-1}{Nn}S^2$ and if you try to replaceS^2 bys^2 which can be estimated in R using the command var variance.

So, this is how I can obtain the estimate of variance and if I want to find out the standard error of the sample proportion under SRSWR, then I have to simply take the square root of this quantity. So, that is what I am writing here this is the estimate of the variance and here I am trying to take the positive square root of the variance of p.

(Refer Slide Time: 23:09)



Estimation of Variance and Standard Error of *p* using R in SRSWR :

In our example, the variance of *p* is estimated by

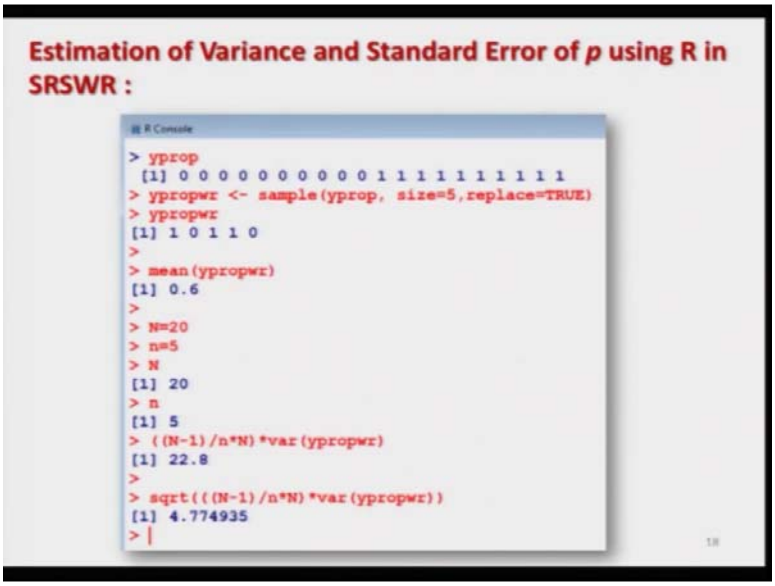
```
> N <- 20
> n <- 5
> ((N-1)/n*N)*var(ypropwr)
[1] 22.8
```

The standard error of *p* is found by the statement

```
> sqrt(((N-n)/n*N)*var(ypropwr))
4.774935
```

Now, I try to execute these things on the sample value that we have obtained. So, we have hereN equal to 20,n equal to 5 and using the command for the variance of sample proportion in the data ypropwr, it comes out to be 22.8 and if you try to take the square root of this quantity, this will give you standard error of sample proportion isp which is here like this right. So, you can see here it is not difficult to find out these things over here and this is the screenshot of the result which I have presented here.

(Refer Slide Time: 23:46)



Estimation of Variance and Standard Error of *p* using R in SRSWR :

```
> yprop
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 1 0 1 1 0
>
> mean(ypropwr)
[1] 0.6
>
> N=20
> n=5
> N
[1] 20
> n
[1] 5
> ((N-1)/n*N)*var(ypropwr)
[1] 22.8
>
> sqrt(((N-1)/n*N)*var(ypropwr))
[1] 4.774935
> |
```

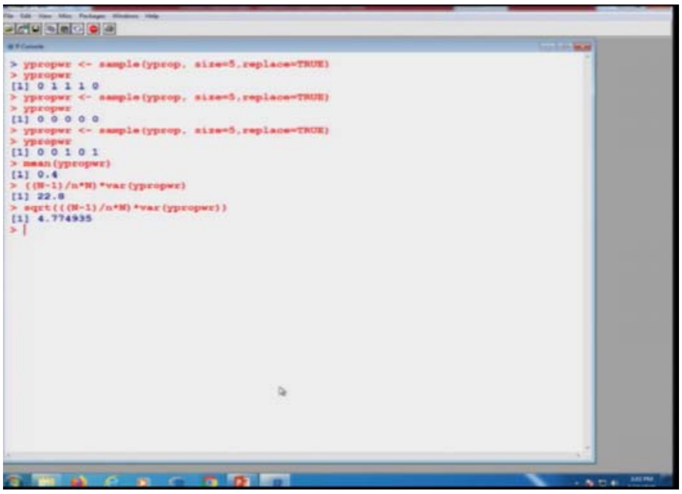
And now, I will try to come to the R console and I will try to show you all the things. So, I already have created the sample, but I will try to use the same command over here. So, I will try to use the sample and then, now I try to find out its mean as well as variance step and standard error, right, ok. So, you can see here that I already had obtained the sample.

So, my sample here is same ypropwor you can see here this is y p r o p w o rletters. So, I would try to operate the commands over this thing. So, if I want to find out the population estimator that the sample proportion to estimate the population proportion so, I simply have to find out the mean of these values.

So, you can see here there are three values here which are 1's so, 1 plus 1 plus 1 3/5 which is 0.6 and if I try to find out the variance so, this is the variance of for sample proportion this will be the data vector here is ypropwor and, but you will see here, you will make a you will get here the value 18.

Why? Because I already had, I can show you that I already had entered the value of hereN which was 20 andn was here 5, right and if you want to find out the standard error so, you simply have to take the square root of this quantity, this comes out to be 4.24.

(Refer Slide Time: 26:26)



```
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 1 1 1 0
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 0 0 0 0
> ypropwr <- sample(yprop, size=5,replace=TRUE)
> ypropwr
[1] 0 0 1 0 1
> mean(ypropwr)
[1] 0.4
> ((N-1)/n*N)*var(ypropwr)
[1] 22.8
> sqrt(((N-1)/n*N)*var(ypropwr))
[1] 4.774935
> |
```

So, now, I come to the case of simple random sampling with replacements. So, you see here I try to first generate the sample here and I have cleared my screen so that you can see it clearly, you can see this sample I have denoted yprop under wr.

You can see here this is 0, triple 1, 0 the sample of size 5 and if I try to repeat this thing, this is again 0, 0, 0, 0, 0 well, you see this is possible because the probability is not 0, but we know that it is not a very nice sample so, let me try to generate one more sample and you can see here this is like here 0, 0, 1, 0, 1.

Now, if you want to estimate the population proportion, you simply have to find out the mean of this sample which is 0.4 and; obviously, there are two units out of 5 which are 1 so, this is 0.4. And similarly, if you want to find out the variance, the command for the variance here is so, I copy the command here and if I execute it, this is 22.8 and if you want to find out the standard error so, you simply have to take its square root, and this will come out to be like this 4.77, right. So, you can see that it is not difficult to compute this thing right.

So, now, I have shown you that how you can estimate the population proportion, its standard error in case of the simple random sampling for qualitative variable and if you want to compute the confidence interval, I had explained you in more detail means I had explained you all the step that how are you going to compute $Z_{\alpha/2}$ using the command q norm and so, I am sure that it is not difficult at all for you to write down the one-line expression for finding out the lower and upper limits of the confidence interval.

So, that I will leave up to you. I already have shown you that how you can computep, how you can estimate the variance of p and in order to find out the value of $Z_{\alpha/2}$, you have to simply find out the percentile which can be obtained for N(0, 1) distribution using the command qnorm qnorm and you can obtain the required confidence interval. But after that, you also have to make the continuity correction by adding and subtracting 1 by 2.

So, now, I will stop here, I have given you all the details that how you can draw the simple random sample for a qualitative variable, for a quantitative variable, how to estimate the population mean population proportion, how to find out the standard error of the sample mean or the sample proportion, how to construct the confidence interval when σ^2 is known, when σ^2 is unknown.

And both type of confidence interval in the case of quantitative variable, the confidence limits are exact whereas, in the case of qualitative variable, the confidence limits are say approximate because that is based on an approximate N(0, 1) distribution. And you also have seen it is not difficult at all to compute them or to estimate them using the R software. So, now you can say the journey starts here for you.

You can imagine that there is some online shopping site which has gathered a population of millions and billions of observations, and they want to know something without going into all the observation because they can challenge the computational power or the structure or the architecture of the computer possibly, they may not handle that many observation.

This sampling gives you a solution that you need not to go for all the values, but you simply have to choose a sample of reasonable size and then, computes compute everything whatever you wanted to compute for the entire population just on the basis of a small sample. So, now this is your turn, think about it, think about the areas where you can use it and after that who says becoming the data science scientist is difficult. Those who say, they do not know what is statistics.

So, you try to learn and on the next lecture, I will come up with a new topic. Till then, goodbye.