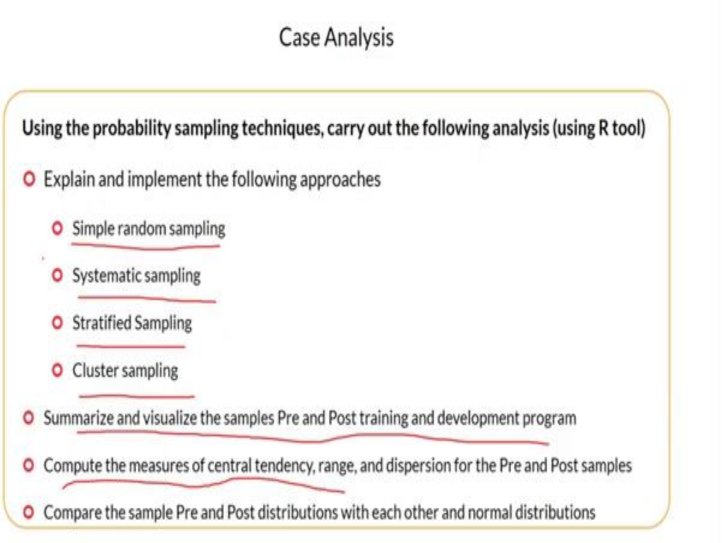
Artificial Intelligence (AI) for Investments

Prof. Abhinava Tripathi
Department of Industrial & Management Engineering
Indian Institute of Technology – Kanpur

Lecture – 22
Case Analysis

(Refer Slide Time: 00:00:16)



Case Analysis

Using the probability sampling techniques, carry out the following analysis (using R tool)

- Explain and implement the following approaches
  - Simple random sampling
  - Systematic sampling
  - Stratified Sampling
  - Cluster sampling
- Summarize and visualize the samples Pre and Post training and development program
- Compute the measures of central tendency, range, and dispersion for the Pre and Post samples
- Compare the sample Pre and Post distributions with each other and normal distributions

Now that we have seen various approaches to examine data, let us start with sampling exercise. Using the probability sampling techniques, we will carry out the following analysis with our tool. First, we will try to understand various approaches to sampling and their implementation in R. Broadly, there are two kinds of approaches; probability sampling approaches which include simple random sampling, systematic sampling, stratified sampling and cluster sampling.

For empirical studies, these approaches are important. In addition, there are non- probability sampling approaches, such as judgment sampling, convenience sampling and snowball sampling. From empirical analysis perspective, these approaches hold less importance. Therefore, we will explain and implement the probability sampling techniques, namely probability sampling, systematic sampling, stratified sampling, cluster sampling and random sampling.

We will start by summarizing the pre and post samples. Next, we will compute various measures of central tendency. These include mean, median, mode, and quantiles. Then we will compute measures of range and dispersion. These include range, variance, standard deviation, and mean absolute deviation. Next, we will visualize the density distribution of pre and post samples and compare them with each other.

We will also compare the pre and post sample density distributions with normal distributions having mean and standard deviation that is same as pre and post sample distributions. **(Video Starts: 00:01:56)** Now that we have discussed the summary and visualization of the data, we will move to the sampling aspect. We will discuss a number of sampling techniques as a part of inferential statistics we analyse and interpret data.

In the previous videos we discussed and computed various summary statistics, descriptive statistics for the Xio Limited case study. In this video we will discuss sampling and it is different types. A population contains all the items or individuals of interest that one seeks to study. A sample contains only a portion of population of interest. One analyses a sample to estimate the characteristic when we choose a sample.

We want it to be as similar as possible to population. There are many ways to collect a sample. Now, we will discuss the sampling techniques one by one. Predominantly there are two sampling techniques, one is probability sampling and there is one non-probability sampling. From empirical perspective probability sampling is the one important. We will discuss a little bit non-probability sampling but it remains less important to the empirical aspects.

In probability sampling you have simple random sampling, systematic sampling, stratified sampling and cluster sampling. For this particular case study, simple random sampling remains the most important as we will see. Let us start with the very basic simple random sampling. We will use the sample command. It is provided in base R data functions. I will apply it on my pre-data to extract the sample of pre-variable.

I will tell it that I need 50 observations from the overall 1000 observations and I will also give a command replace equal to T which tells R that it should be done with replacement that means the observation taken in the first sampling should be replaced back in the original data. Now, one way to do this is to simply run this command and you would notice that certain output has appeared on my console.

I can run this command again and you notice a different number of variables on my output. This has happened because every time you extract the random sample R changes the sample. So, one way to do that is to run this command called set.seed and establish a seed number. Let us say, set.seed as 51. Once I do that and if I run, notice that if I run the sample command notice a certain set of sample has appeared.

If I now, if that I have run the set.seed command I have fixed it. So, every time I run set.seed at 51 you notice same sample is appearing which is needed to generate the same sample. Now, a more probably a different and more relevant way in this context is to use a variable name pre and assign the sample generated to this pre-variable. I run this command where I assign the sample to my pre-variable.

Similarly, I run a very similar command to assign the post variable sample. So now, I am assigning the sampled values to my pre and post variable and they are ready for further analysis. For example, I can start with a basic summary operations. I can summarize the data how I can use simple summary commands to summarize previous sample. Notice, the minimum, maximum, median and other inferential statistics and also notice when I run it on post the measures of simple tendency are appearing.

And we can compare the summary of pre-sample as well as post sample. Now, please notice there is a clear shift in the pre-sample versus post sample and more importantly please notice here using the sample not the entire 1000 observations but just a small sample from it I also reached the same conclusion. So that shows the importance of samples in making inferences about the original data.

We will also check the mode of this data, as we did earlier for that we need that library statip that we earlier installed. We will use that library statip and now, we will see some of the initial most frequent observations or that are part of it is mode for pre-variable. Let us see some of the initial observations. Similarly, we will use the post variable, for its most frequent observations.

Please noted that in console once I run this command clearly, there is a shift in most frequent observations from pre to post. That indicates that there is indeed some impact of training. Next, we can see some of the other measures of central tendency and range as we have seen. So, for example, we can look at the quantile measure so, start with quantile command. I can run the quantile command on my pre-variable and I can see this for 20 percentile intervals, with this sequence.

I can specify to R that starting from 0 to 1, a 20 percentile interval is to be produced. I can do the same for my post variable also, as you will notice here. And please notice all these measures on sample are indicating a shift I repeat, a shift on the over original data. So that is interesting because we are able to make these inferences using our small sample of 50 observations.

So now, that we have discussed summarizing and computing measures of central tendency for a sample of 50 observations which we extracted using simple random sampling. We will move towards more measures of range and variability after this. Now that we have understood extracting random sample from the data and we have also seen how to compute measures of central tendency for this data.

We will start with computing measures of range and visualizing the data. So, we will start with computing range measures and measures of dispersion. So, for range measure, a very simple formula range can be applied to the pre variable as well as the post variable. We can clearly see the pre and post samples there is a difference. The range of pre for both lower and higher values is on the lower side.

That means the range has shifted after the training and development program which suggests that there is some significant impact of training and development on employee satisfaction. We will also check the variance of pre-sample and variance of post sample the variance of pre and post sample are similar. We can also compare the standard deviation of pre-sample and standard deviation of post sample.

There appear to be also quite simple, similar. If you remember the population measures or the data when we are referring to population, we mean to say data only. So, the data these measures were quite similar. That means the inferences that we are drawing using the sample are quite similar to the ones we drew when we were working on the original data or the population.

That means inference is drawn from samples may give some indication about the nature of original data. We can also compute mean absolute deviation that is a MAD measure for pre and mean absolute deviation that is MAD measure for post-date sample data also. Although they appear to be slightly different. Now that if you understood the computation of range and dispersion measures, let us focus on visualization of this sample.

First, we will start by plotting the density plot of pre-measure. For that we will again use the plot command on density of pre-variable. We will add the main heading central heading as density graph. The x-axis label, x lab argument can be set as data we can use red colour col argument can be set as red. Also, we can set the line with argument lwd as 4. As seen earlier we can also describe the x-axis limits as 2 to 7.

And we can also set the y-axis limit with ylim argument as 0 to 0.8. Now, let us plot this data we will do an interesting exercise now, we will try to superimpose a random normal distribution which has the same mean and standard deviation as this pre-sample measure. Let us do that so, first and foremost, we generate a random normal sample. We will name it Norm_Pre and we will generate it with the following command rnorm, 1000 values we will give to the sample.

And a mean which is same as mean of pre-sample data and the standard deviation which is also similar to standard deviation of pre-sample data. Now, we have generated the sample, we will plot the density of this sample with the lines command. We will not draw a new plot we will use the lines command to draw the density of this plot on the same original old plot. We will give it a colour of green and we will set the line width as 4.

Let us see how it appears and interesting thing appears we can see that the distribution, the density distribution of pre-sample data and a normal distribution with same mean and standard deviation are not similar. While they may appear to be symmetric to us but their peak is quite different. The random normal distribution has theoretically well justified peak. While, if you look at the pre-sample data, it appears to be bimodal.

So, there are two peaks that means two set of observations have particularly large probability densities. With this in a very similar manner, we will also compare the density plot of post data with a random normal distribution which has the same mean and standard deviation as the post data. So, the first step we will plot the post data. Again, we will use the same convention, we will use the blue colour. So, we will plot the post data.

We will generate a random normal variable, with name as norm underscore post which has the 1000 values but mean and standard deviation same as post sample data. So, we will plot this or superimpose this normal distribution which has a mean and standard deviation same as post sample data will plot this. Notice again both of them, the post sample data and random normal distribution may appear similar in terms of their symmetry but again their peaks are quite different.

As expected, the random normal distribution has quite a nice peak as expected from theory. But the post sample data again appears to be slightly bimodal with two different peaks as it appears. And both of these set of observations appears to have quite high probability density and it appeared to be sort of bimodal distribution so, not similar. But please remember our discussion about central limit theorem.

Even though this post sample and pre-sample data are not or do not conform exactly to normal distribution. But in large samples repeated sampling, often it is argued that the statistics computed from the sample approaches to normal distribution. And therefore, CLT central limit theorem allows us to do various statistical procedures with assumption of normality.

Now, let us superimpose the pre-sample and post sample data with each other. We will try to compare them, we will compare pre-sample and post sample data we have 50 observations of these data. So, first as a first step, we will plot the density of Pre variable again like we did earlier. So, we will plot the using the same command we will plot the density of pre-variable. Now, we will plot the or we can say superimpose the density of Post variable on the Pre-variable.

So, as a first step, we are plotting the pre sample then post sample and then we can also add a legend to it. Legend can be easily added on the top left corner. Maybe we can decide the location as we may find suitable. We can again, as earlier we can give the name post and pre to post and pre-sample data. We will follow the earlier convention of colour. We will give the fill at red for pre, blue for post, so, we will fix the colours and then we will add the legend.

Now again notice both of these samples give a clear picture that there is a shift from pre to post due to training. Now, this inference is quite similar as we draw that from our original data or what we are calling as population. The inference from sample and population are quite same that is in the random sample we clearly find that there is a shift in distribution. Interesting to note that without using all that 1000 observations from the original data.

We are able to make the same inference this with just 50 observation sample. With this we have understood the random sampling we will also try to examine some of the other methods of sampling, as we discussed earlier. Now that we have understood simple random sampling, we will start with other types of probability sampling, starting with systematic sampling. As part of systematic sampling individuals are chosen at fixed intervals, from the population data.

For example, to create a sample of n from a population size p with fixed interval k you have k = p/n. That means, if you have population of 1000 and you want to create a sample of 50, you need to pick observations from every 20 observation sample. So, your length of observations is 20 and you will be able to choose 50 observations. Now, why would you do that?

Many times, if you believe that your samples are coming from a fairly uniform observation, so that if you divide your observations in a certain intervals let us say 20, 20 and you have 50 intervals the samples that you will get from each of the 20 observations are not so different from each other. And they fairly represent your population then it will save a lot of time and resources while doing the sampling exercise and you would be comfortable fixing a certain length.

Now fixing the length is a trade-off. If you increase the length of interval, you will get less observations in your sample. If you decrease the length, you will get a large number of sample but again the time and cost involved in sampling may be higher. So, as a starting point, first we need to install a package called TeachingSampling. So, we will install this package called TeachingSampling.

Once you install this package, you need to put this in your current working library. This package TeachingSampling, will put it under the current working directory. Now, the format the syntax to implement is quite easy. We will use this variable Syst_sample = S.SY (1000,20), we will tell R that there are 1000 observations and we need to break it into 20 intervals this will generate a vector of indices.

This vector if I want to check the length of this vector, a vector of indices, has a length of 50 and inside this index vector we have locations. I repeat we have locations starting from 1 to 1000 in between we have randomly chosen samples which indicate the location of variables. So, if I want to select the location for pre-sample, let us say we select for pre-sample.

```
Syst_Sample = S.SY(1000,20)
Pre = Data$Pre[Syst_Sample]
```

The location is equal to we can choose our variable data dollar pre and we can select the index variable as our location variable by putting it inside the subscripts for data pre-variable. And we can generate a set of 50 observations which are systematically chosen observations for our pre-variable. Similarly, I can select another set of 50 observations, purely randomly chosen for our post variable as well.

```
Syst_Sample = S.SY(1000,20)
Pre = Data$Post[Syst_Sample]
```

So, I can again run the same set of codes for our post variable also and we can generate our post variable. Now, we can do all the statistical analysis that we have done for our random sampling exercise. We can repeat all the exercise again but just for saving time we will not do that but we will simply only see the summary of our newly created period post samples. So, just have a look at our newly created pre and post samples.

And please notice there is a clear cut shift notice that there is a clear shift in our pre and post sample data which suggests that indeed there is an impact of training on our pre and post sample. Now, let us start with the stratified sampling. It is another part of probability samples, so, we will talk about stratified sampling. In stratified sampling you want to create or you already have some predefined stratas in your data and you do not want to ignore these stratas.

For example, you may have in your survey, respondents male and female candidates and you would not want to miss the responses from one set of gender. For example, you would want to give certain equal weightage in your both of these stratas of male and female respondents. So, you would try to take a sizable component or sizable number from both of these status. Let us do that for this particular example.

So, for that we need to install dplyr package to implement the stratified sampling. We need to install dplyr package once it installed we will add this dplyr package to our current working library. Now, will simulate the gender variable where female = 1 and male = 0. In this way we will simulate the gender variable for that we can simply add a gender variable, a simulated variable to our data we will define this as factor.

Factor is a categorical variable in R. So, the procedure to generate the simulated gender variable is quite simple. We will run this runif command and generate 1000 values from 0 to 1 and will round them so that we have exactly 1, 0 values. Since we have converted them to factor they are meaning as a numeric is lost. They are only factor to R now so, our gender variable is created.

```
Data$Gender = as.factor(round(runif(1000,0,1),0))
```

Once we have this gender variable, we can run a sample command by going inside the data variable. Inside data we group it with group_by function and we group according to their gender. Inside gender we apply our sample command and from each of the genders we pick exactly 25 observations. This will help us giving equal representation to both male and female respondents, so that is 25 are chosen from male and 25 are chosen from female gender.

```
Sample=Data%>%group_by(Gender)%>%sample_n(.,25)
```

We will run this command and now, we will extract the pre and post sample data, for example, with this command. We will extract our pre-sample data and with Sample$Post, will extract our post sample data. Now, again, we can repeat all the summary and visualization analysis that we have seen for our earlier samples. But just for the sake of it we will add summary measure for pre and we will also add the summary measure for post.

We have given equal representation to both of our strata that is male and female. And notice there is a clear shift, clear-cut shift from pre-sample to post sample, so, this is our stratified sampling. Now, we will talk about cluster sampling. In a cluster sampling approach the population is first divided into small groups, known as samples. Then randomly we choose a certain number of clusters and then, once you have chosen certain number of clusters.

You probably choose observations inside each cluster. The business argument behind choosing clusters can be like this. For example, you may have different, different income segments in each department of your organization. Now, you would not want to miss out on any particular income segment. Probably you would want to give equal representation to each income segment.

So, therefore, you cluster in each department, you create different, different income segments. Each department will have certain high income segment and certain low income segment. Now, you can select, maybe from a large number of high income segment. You can select a certain high income segment. Again, repeat the same for low income segment and again, if there is a another category called mid income segment.

You also pick a certain clusters from mid income segment. Now, from each of these clustered segments, you can sample data. This will save a lot of time and cost for your sampling exercise. You need not look at all the clusters. You can probably from homogeneous clusters, pick and choose a smaller number of clusters. So, this is the kind of sick business situation where clustering exercise or cluster sampling would help how to implement that in R.

So, for that we need to install a package called sampling. Once this package is installed, we need to add this package to our current working library. Now that this sampling library is added, we will first and foremost generate a sequence of department, a simulated number from 1 to 50 for all the 1000 observations. So, we are adding a sequence and defining it as a factor because it is representing a department.

So, each number corresponds to a department. It does not have exactly a numeric meaning it is more of a categorical variable. So, we are generating a sequence of 1000 numbers. Randomly assigning numbers from 1 to 50 each number from 1 to 50 represent one of the 50 departments. So, once we have generated you can see also, so, this will be a sequence of 1000 numbers, categorical values, starting from 1 to 50, these are departments for us.

```
Data$Dept = as.factor(round(runif(1000,1,50),0))
```

Now, we will do our clustering exercise. So, we will assign these clusters to a new variable called sample, wherein we will run our cluster command on our data variable called data will give a cluster name equal to department. So, a cluster of departments will be created. Now, we will give a size of 10 that means out of 50, only 10 departments will be chosen. Probably our belief system suggests that all these 50 departments, 10 departments are fairly representative.

So, also we are using a technique called simple random sampling without replacement srswor for our clusting procedure. The idea is that the probability of each of this cluster of being selected is same, so, there is no, it is without replacement. So now, let us see the dimension of this cluster variable. Let us see the dimension of this data sample. You have created this data sample now, let us find the dimension of this data sample it is around 216.

```
Data_Sample = cluster(Data, clustername=c(“Dept”), size=10, method= “srswor”)
dim(Data_Sample)
levels(Data_Sample$Dept)
```

We can also see what are these departments that are selected for our exercise? So, we can see that we can add this levels command and once we add this levels command to our data sample dollar department. Notice that it shows me all the 50 departments. However, we have asked the command to select only 10 samples. The problem here is that as the way R works it keeps or the original cluster of original departments are still stuck to the variables.

```
Data_Sample$Dept=droplevels(Data_Sample$Dept)
Levels(Data_Sample$Dept)
```

So, how do we remove this? We add a very simple command called drop levels on our original data. So, we will add, drop levels command and drop these unused levels from my data once you do that those levels are dropped. And now you, when you run the levels command notice, only 10 levels are appearing these are the representative levels that are chosen.

Once we have selected the levels or departments the 10 departments we can very well select a certain number of observations to get our final sample. So, we will name it final sample which is equal to Data_sample. So, we are selecting the data from our original sample with 10 departments and we are grouping it by department only. Inside departments from each department we are using the techniques of sampling.

And from each department we are picking only a size of 5 sample. So now, this final sample, let us see the dimension of this final sample, so, there are only 50 observations as we would have wanted. Now, let us say select the pre-sample variables from this final sample let us select that. And we will extract the pre-variable first. So, we will first extract the data, we will use this getdata command.

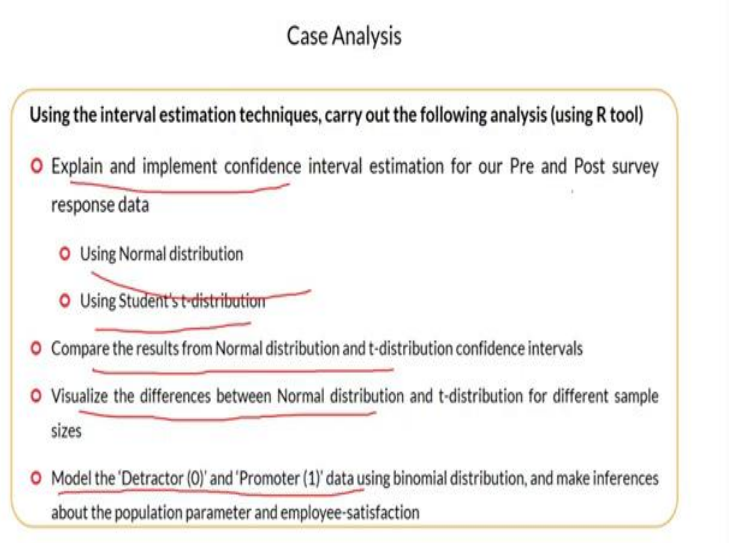
```
Final_Sample=Data_Sample%>%group_by(Dept)%>%sample_n(size=5)
dim(Final_sample)
```

And from the original data which is my data variable I will extract this final sample and we will name it new data. From this new data we will extract the pre-variable which is of importance to us pre-variable we will assign it. Similarly, we will extract our post variable also from this new sample, while other visualization and summary analysis remains the same as for the random sampling that we did.

```
New_Data=getdata(Data, Final_Sample)
Pre=New_Data$Pre
Post=New_Data$Post
ummary(Pre)
```

Let us only look at the summary variable for our new pre-variable obtained using cluster sampling and also, we will do the same for our post variables. Again, please notice, as we have seen earlier as well there is a clear-cut shift in the pre-sample and post sample values indicating that there is indeed a significant impact of training and development exercise on employee satisfaction.

(Refer Slide Time: 00:49:13)



Case Analysis

Using the interval estimation techniques, carry out the following analysis (using R tool)

- Explain and implement confidence interval estimation for our Pre and Post survey response data
  - Using Normal distribution
  - Using Student's t-distribution
- Compare the results from Normal distribution and t-distribution confidence intervals
- Visualize the differences between Normal distribution and t-distribution for different sample sizes
- Model the 'Detractor (0)' and 'Promoter (1)' data using binomial distribution, and make inferences about the population parameter and employee-satisfaction

Now that we have seen various approaches to sampling the data, let us start with confidence interval estimation. We will explain and implement confidence interval estimations for our pre and post survey response data. For this we will use normal distribution with z statistics and students t distribution with these statistics. We will discuss the implementation in r in a detailed manner, to explain the concept and also see the quip implementation as well.

We will compare the t and z values and also the resulting confidence intervals. We will try to describe and explain the differences between both of these distributions and the implication for analysis will also simulate normal distribution and student's t distribution to understand the difference and it is implications for confidence interval analysis. Next, we will also discuss a case of binary distribution.

We will discuss a business case wherein the organization focuses only on certain individuals with a score of less than 4.5 as detectors and more than 4.5 as promoters. The organization wants to focus their training and development programs on these detectors with a score of less than 4.5. Therefore, it assigns a value of 0 to these detectors and 1 to promote us. Given the binary nature of data that is 0, 1 categories it needs to be modelled using binomial distribution.

And using confidence interval estimation we need to assess whether the data fairly describes the original data with 1000 observations, we will also try to make some inferences about the impact of training and development, pre and post sample data. And further make inferences about this impact on population. **(Video Starts: 00:51:04)** Now that we have understood various sampling techniques.

We will employ them in our confidence interval estimation and we will understand this process of confidence interval estimation and it is application with r. So, we will start with confidence interval estimation. When we wanted to estimate the population parameter. Let us say, population mean, beam can employ sample estimator. For example, sample mean, in fact if you want to estimate any population parameter, you can use sample statistic.

So, in this fashion, you define confidence interval as an interval surrounding the parameter and the interval has a certain chance of being the true statement or definition of being the population parameter. So, how do we interpret this confidence interval? The statistical interpretation is quite simple. It is the confidence interval, it has a certain probability 1 – alpha where alpha is the critical value or the complement to the confidence level.

This probability of containing the population parameter. For example, if you have 95 percent confidence interval which falls between 0.65 to 0.73. Then you would say that there is a 95 percent chance that the interval of 0.65 to 0.73 will contain the true population parameter. Now, we will perform this exercise on our pre and post sample data and we will do this in the step-by-step manner.

Please note there are certain differences between t distribution and z distribution. We will use these t statistics and z statistics n our interval estimation. There is a fundamental difference between these two distributions. The t distribution depends upon the degrees of freedom, as we have discussed and curves with more degrees of freedom, are taller and have thinner tails, t distributions have heavier test than the z distribution.

So that means it needs more evidence or heavier values of t are required for same level of confidence interval estimate limits. So, let us start with our interval estimation process and first we will install the relevant packages. In this case it is distribution 3, this is our relevant package. We will install this package and once we have installed this package we will use library command to put this package in our current working directory working library.

Now, first, we will start with the z test statistics. Although z test statistic is less spectacle, as we have discussed, t statistics is employed when either you do not know the population standard deviation or in cases where you have less sample value, for example, 30 40, 50, values in your sample. In those cases, you prefer to use t statistic which is more efficient in these cases.

So, but first we will start with z test statistic and confidence interval estimation. What we will do is we will try to estimate some population parameter like population mean using the sample estimates. Conduct the confidence interval estimation and see whether actual population parameter is falling in that interval or not. That will give us some idea about the efficiency and effectiveness of the sampling process.

So, in our case we have already taken the sample from our random sampling process. The sample mean is 4.05 but this is just a sample estimate. We need to generate the confidence interval to say that actual two population parameter lies in that interval. So, let us decide our confidence interval as 95 percent. So, our complement value or critical value is 0.05 that is 5 percent. And therefore, obviously our confidence interval is 95 percent as you can see.

Next, we need to generate a normal distribution, since it is z statistic based confidence interval estimation, we need to have a normal distribution. In fact, we need standard normal distribution which has a mean of 0 and standard deviation of 1. So, now that we have generated this normal distribution, let us look at some of the parameters. So, if you want to check the mean of this normal distribution it is 0.

The standard deviation of this normal distribution is 1. In fact, we can see the plot and also make out some inferences about its characteristics. So, when I plot this, it clearly appears as a very normal, smooth, stable normal, standard normal distribution. As a second parameter we also need the standard error of estimate. So, standard of the estimate in case of z statistics, it is standard deviation of the population which would be Data$Pre.

Since in this case the 1000 observations are used to generate the small sample of 50 observations. We will consider this original data as population. Only so, it is standard deviation is employed in computation of standard error divided by the square root of sample size. So, sample size in this case is 50 which can be easily computed as length pre. So now, we have our standard error of the estimator which is around 0.08.

Next, we also need to compute the value of z statistic, z statistic can be very easily computed with the help of quantile command. I can generate the relevant quantile for our normal distribution which we have just generated and specify the confidence interval that is 1 – a by 2, why 1 – a by 2 we are using? Because we are saying it is a two-tailed test. That means the actual population parameter, can move either on the upside or on the lower side both positive and negative side.

So, we are considering both tails and therefore, we are using 1 – alpha by 2. That means starting from 2.5 percentile to 97.5 percentile. Since our confidence interval is 95 percentile. So that means this 1 – alpha by 2 will represent 2.5 percent to 97.5 percent interval which is essentially 95 percent only. Now, we have computed our z value in this fashion. Let us compute our margin of error MOE margin of error = z statistic into standard error.

Now we are done. We need to compute the lower limit of our confidence interval very easily computed as mean which is our sample estimate of mean – margin of error. Similarly. I can compute the upper value of the confidence interval very easily computed mean of sample + the margin of error. So, if I compute these values like, let us have a look at these values. So, the lower value is 3.90 rounding off and the upper value is very easily can be seen here 4.21.

So, this is my confidence interval of 95 percent, where I am hype where I am estimating that in this interval I have 95 percent confidence that my true population parameter should lie. Let us look at the true population parameter value. In this case, we are saying that original data is data itself and we are assuming it to be our population I can simply compute, it is mean as 4.00.

So, it appears indeed, this population parameter falls inside that 95 percent confidence interval. So, our sample is reasonably good. Now, as you keep on increasing the level of confidence, you increase the sample size and decrease the alpha value. It does a trade-off. The more confident you want to be. You can increase the confidence interval but at the same time there is a cost in terms of efficiency.

So, either you increase the confidence, interval and decrease the alpha value or you increase the alpha value and decrease your confidence interval. So, with this we have been able to compute the interval estimation using z statistic. Now, in the next video we will talk about confidence interval estimation using t statistic. In this video we will perform confidence interval estimation using t statistic.

So, we will have confidence interval estimation using t statistic, t statistics are defined by their degrees of freedom corresponding to the sample. So, first we need to compute the degrees of freedom which can be very easily computed with this length command. I can compute the length of my pre-sample which we have been working on and this D of is my degrees of freedom.

Next, I need to compute the t-statistics which is quite easy. I can use this t = qt function, qt function is basically to generate the quantiles from t distribution. So, I will use this and my critical value I will define as alpha by 2 because it is a two-tailed test. So, as we discussed earlier, we are using alpha by 2. Then I assign DOF degrees of freedom to my df variable df = DOF. And I specify that lower.tail = F.

If I do not do that it will give me other till, for example, if it is a 95 percent confidence interval. If I put a lower.tail as true, it will give me 2.5 percentile value. If I put it as false, it will give me 97.5. So, both cases are fine but in this case I am using lower.tail equal to false. As a next step I will compute standard error of the estimate. Next, set of steps are quite similar to what we have done earlier?

So, standard error equal to standard deviation of pre. Please notice that in the previous case, we were using the population standard deviation but as per the assumptions of t-statistics. We already know the con the variation of the population in case of z-statistic. But in case of t-statistic, it assume that samples are small. And generally, we do not know the variation or standard deviation of r population.

So, with this I divided by square root of length of pre variable and in this fashion I compute my standard error of the estimate. Next, I compute margin of error as done earlier, margin of error is t into SE which is my standard error. So, I get my margin of error. Then using this margin of error, I can compute the lower bound of my confidence interval which is mean pre – MOE. And I can compute the upper bound as well.

So, we can see this confidence interval easily so, lower is 3.88 and upper is 4.23. Now, if I want to confirm whether my true population parameter which is the mean of the sample that not the sample. But the actual data original data itself that is Data$Pre, let us see that. So, my mean of original data is 4.00 which actually, false in this confidence interval. That means I my confidence interval where I said that with 95 percent of probably t.

I assume that micro population parameter should be lying in this range which is true it seems. But please also remember that when we computed this exercise through z-statistic. Then also we successfully found that our true population parameter lied on the same confident interval using z-statistic as well. And therefore, whatever the need to conduct this exercise using t-statistic.

And for that please have a look at t and z values computed in this size. T value is 2.0 and z value is 1.96. This is a clear indication that for same level of confidence for me to have the same level of confidence, I need a relatively higher evidence. I need a higher t value to generate the same level of confidence. To talk about more on this let us try to visualize, let us try to visualize what is happening here?

So, we will plot the density plot of the normal distribution using rnorm command. And we will generate, let us say 50 maybe 100 values. I will assign a colour of red, line width of maybe 2. And will not sign any central heading here that we will do in legends. We will add legends to make it more clear. The plot is provided on the plot window and now I am adding the t-statistics also.

And for that I need to use rt command which will generate the density distribution for a p distribution with 500 observations and let us put df as = 20. Again I will use the same argument colour I will put to green. And remaining arguments remain the same. So, when I generate this graph, please notice. The t-statistics is very clearly very different from this one the red one. Let me add, legends also to make it more clear.

So, I will add legend also the legend to be added on top left side. I will give the name as t distribution and normal. I will also follow the same convention by providing the fill as equal to green for t distribution and red for normal distribution. So now, you have the commands and now, I can run this command. So, I have my legends also. Now, please notice as of now, the t distribution is quite different from the normal.

In fact, it deviates a lot notice, I will zoom it notice the tails of this t distribution. The tails are much fatter, while it is peak is lower. The tails are fatter which indicates that for a t distribution, it has higher probability of finding observations on rather extreme tails. And a lower probability at closer to me. That means it is mean and more are smaller which suggests that the most frequent observations that are closer to mean or others.

And measures of central tendency like median mode is much lower and more observations are spread much further away from these measures of central tendency. With this understanding, let us change some of these values. Let us generate slightly higher values, as you can see, as I increase sample size both of these distributions. They become closer to each other.

For example, if I put let us say, 5000 observations and increase my degrees of freedom to 200 very large number of observations. They almost appear identical which means when I have large number of values, whether I do it with z-statistic or with t-statistic. The interval estimation process is almost similar but if I have less number of observations, let us say 100. And very low degrees of freedom let us say 10.

And if I run this notice, the plots deviate a lot, the tails are much fatter and the peak is much lower. And therefore, in this case it would be better to perform this interval estimation with the help of t-statistic. So, this is the reason we use t-statistic when we have small number of values. Also, there are some shortcut ways to conduct this teen test statistic related interval estimation.

We can simply use this t.test on my pre variable and notice on the console window. You notice the confidence interval which is same as we obtained, using the detail procedure where we manually computed the interval. Another shortcut to do this is from linear regression models. So, I run this con in command on my lm mod command lm is basically for linear model or regression model.

So, I can regress my pre variable on a constant term that is one and generate the confidence interval for the constant estimates. Please notice the estimates are identical as our earlier confidence interval which is 3.88 and 4.23. So, it is another trick to estimate the confidence interval for the sample or you can use that t.test. So, with this we realize what is the importance of conducting t test with small samples?

And in the same fashion, we can also conduct confidence inter estimation for our post sample variable as well to see the effectiveness of our sampling system exercise. With the sample estimates we can check and compare them and their confidence interval we can construct. And check if the population parameter is falling in those confidence intervals. And therefore, how effective are sample is?

In this video, we will model the data with the help of binomial distribution. Will conduct confidence interval estimation and make inferences about the population parameter and also the impact of training and development on the employee satisfaction. So, we are analyzing binomial distribution and generating confidence intervals. The binomial distribution model deals with finding the probability of success of an event which has only two possible outcomes in a series of experiments that is success and failure.

The business case for our exercise is as follows. The organization wants to specifically focus on those individuals that score low on satisfaction. And wants to design training and development programs that are more suited and customized to their needs. So, an individual below a score of 4.5 will be assigned a value of 0 and is considered as detector. While those above 4.5 will be assigned a value of 1 and are considered as promoters.

So, as a starting point, let us examine the pre sample data. So, for that we will create a Pre_Bino variable this. This variable will be added to the pre sample data. So, we ascribe to a pre sample a value of 0. If it is less than 4.5 through this command and the value of 1 if it is more than 4.5. And this Pre_Bino variable will be added to those employee codes.

Let us have a look at the sum of this variable Pre_Bino. So, this variable suggests that there are 17 employees in our sample that have received a score of 4.5. And therefore, this Pre_Bino variable is 1 more than 4.5 sorry and remaining score of 0. So, if I want to know the fraction of those that have scored 1. I can simply divide this summation by the length of 3 which is 50. This means that 34 percent of the employees in the sample are satisfied.

Now, you may want to check the confidence interval. That is maybe 95 percent confidence interval in which you believe your population parameter will be there. So, for that a simple command prop.test we can apply and we can define the sum. So, this will give us the number of occurrences which is 17, 17 time we have employees that are satisfied. And the total length of sample will put the formula although length is 50.

But still, we will use this formula to define the length. Then we will put the confidence level as 95 percent. If I run this command notice, the confidence interval estimate of the population parameter. It is from 0.215 to 0.488. And the sample estimate, as we already know 0.34 which is 17 upon 50. Now, you would like to know how efficient is this estimate? How correctly it defines the reality?

So, in our case first, we need to know how many times in our population parameter the value is more than 4.5 or less than that. So, for that again we will perform the same procedure but this time we will use the population. We are calling our 1000 observation data as population in this case. So, let us use that from which actually, the random sample was taken. If it is less than 0.5 then we take a value of 0. If it is more than 0.5, we take a value of 1.

So, this is our population parameter pre training. Now, we can compute the summation that is those who are satisfied with very this simple command population pre. And we will divide it by length. So, it is quite simple with this let us see the number, so, it is 25.5 percent which is slightly different from the sample but it is still false in that 95 percent confidence interval. That means our sample fairly accurately describes the reality.

And it suggests that the original population parameter false within the range of 21.5 percent and 48.85 which actually, is true. So, our sample is a fair representation of reality. Now, with this in the next video, we will talk about the post sample, it is properties and whether it is able to provide that confidence interval for population estimate. And will also compare the impact of training and development on pre and post samples.

Now, we will examine our post sample data with the help of binomial distribution and conduct confidence intervals. So, we will examine the post sample data with binomial distribution and confidence intervals. So, first, as we did earlier, we will generate this Post_Bino data. And here in our sample those values that are less than 4.5 will assign a 0 for those values. And 1 for those values that are more than 4.5.

We will assign these values now. If you want to know, how many individuals this value was 1 I can assign I can check the summation Post_Bino. It seems this number is 39. It has increased significantly I will divide it with the length of our sample which is length of post. And notice the fraction this is 78 percent is our sample estimate. So, this is a sample estimate for getting a value of 1. That is response of more than 4.5.

Now, you would like to generate the confidence interval for your population estimate. And as we did earlier, we will use this prop.test sum Post_Bino which will give us the number of values that are more than 4.5 and then sample size. This is post and we will assign a confidence level of 95 percent. So, when I conduct this confidence interval estimation, please have a look at this confidence interval of the population estimate which is 0.64 to 0.88.

And the sample estimate is 0.78 which was expected, as we already saw. Now, to check the efficiency of this sample or the ability of sample to correctly define and describe the population parameters, let us compute the population estimate. And for that I need to generate the values of zeros and one. Again, same procedure I will for Data$Post. In this case, data variable original 1000 survey sponsors access our population data.

Because from this data only we have generated the sample estimates. So, if this is less than 4.5, we will assign a value of 0 and if it is more than 4.5 we will assign the value of 1. Let us compute the values. And now, we will see what is the total fraction of the population post divided by the length of population which is 0.75. So, the sample estimate is 0.78, while the population estimate is 0.75 which is fairly closer.

And also, this number false well within this 95 percent confidence interval. So, first we got an idea that our samples are fairly describing the population. But more importantly, have a look at the confidence interval estimate before and after that is pre and post. So, before the estimate is 0.21 to 0.488. So, this is our estimate before training. After training our estimate is 0.64 to 0.88 which clearly indicates that there is a shift in the employee satisfaction.

So, pre survey sample data and post survey sample data clearly indicate that there is a shift and there is a significant impact of training on employee satisfaction. That means there are much more employees that are shift satisfied with the score of 4.5. More than that that is one very important inference that we can draw from this analysis. And therefore, it seems that our training and development program is successful.

So, this is how we tested. We model the data sample data using binomial distribution. And we conducted the confidence interval analysis and also saw whether our sample properly describes the original population data. And it also successfully indicated the impact of training and development program on the entire population. **(Video Ends: 01:18:28)**