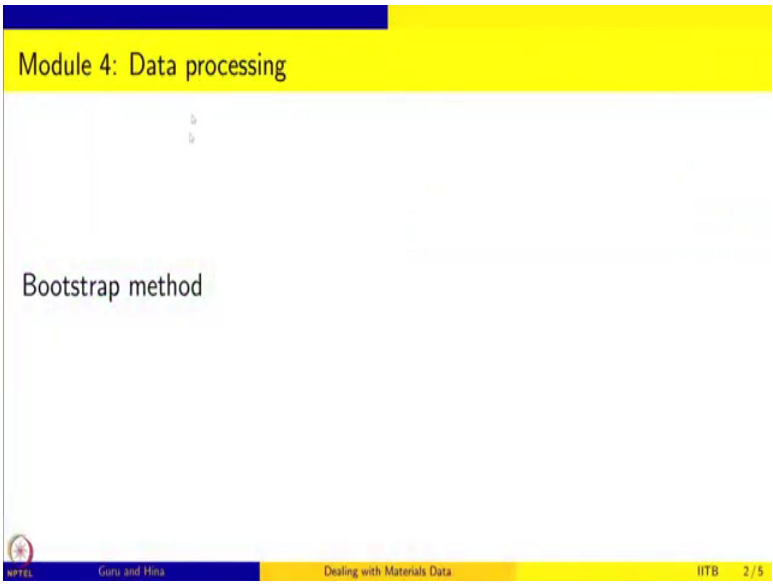
Dealing with Materials Data: Collection, Analysis and Interpretation

Professor Dr. M P Gururajan
Professor Hina A Gokhale
Department of Metallurgical Engineering and Materials Science
Indian Institute of Technology, Bombay

Lecture 66 - Bootstrap method

Welcome to Dealing with Materials Data, we are looking at the collection, analysis and interpretation of data from material science and engineering. And we are in the module on data processing.

(Refer Slide Time: 0:25)

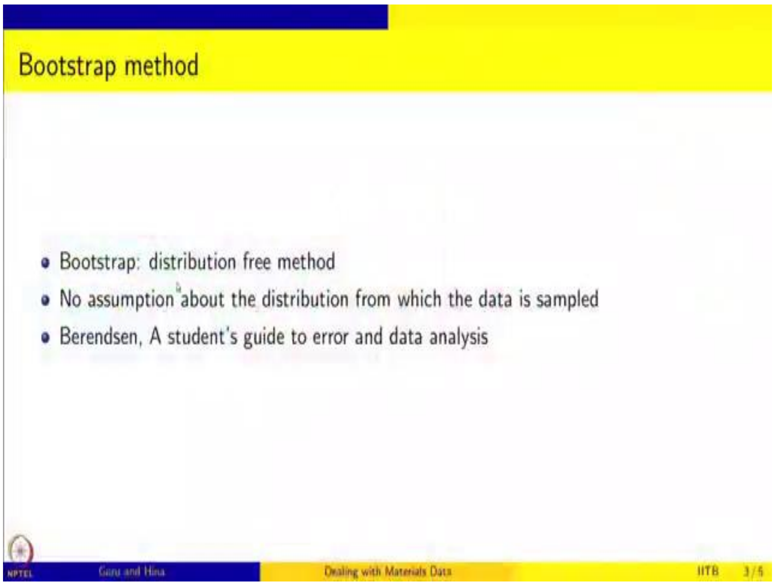


Module 4: Data processing

Bootstrap method

And in this session, we are going to talk about bootstrap method, which is one of the methods to get the accuracy or estimate of the quantities without actually knowing anything about the probability distribution, because in the other methods where we have data and if we know what probability distribution it comes from, then you can give better estimates for intervals and so on. But if suppose you do not know anything about the underlying probability distribution and you cannot make any assumptions, is there a way to get an estimate, interval estimate? And that is done using the bootstrap method.

(Refer Slide Time: 1:07)



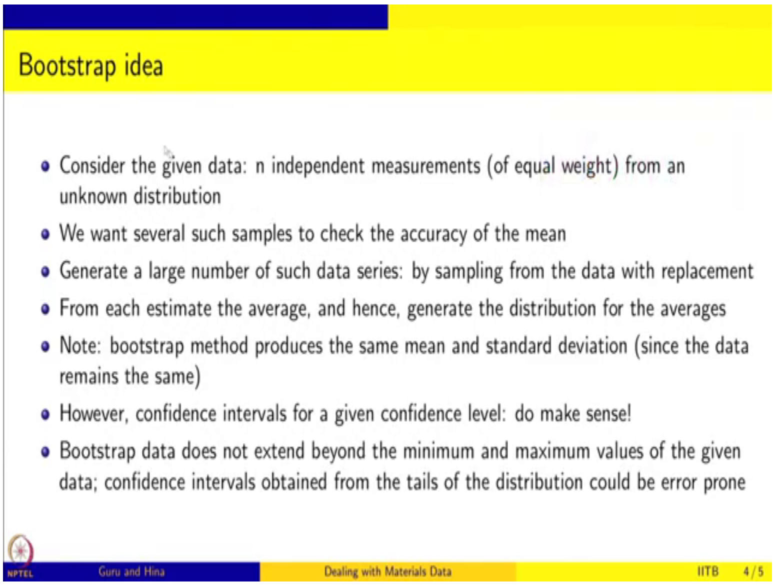
Bootstrap method

- Bootstrap: distribution free method
- No assumption about the distribution from which the data is sampled
- Berendsen, A student's guide to error and data analysis

NPTEL Guru and Hina Dealing with Materials Data IITB 3 / 5

It is a distribution free method. So, we are not making any assumption about the distribution from which that data is sampled. Berendsen in the student's guide to error and data analysis gives details of the bootstrap method, which I strongly recommend that you go through.

(Refer Slide Time: 1:22)



Bootstrap idea

- Consider the given data: n independent measurements (of equal weight) from an unknown distribution
- We want several such samples to check the accuracy of the mean
- Generate a large number of such data series: by sampling from the data with replacement
- From each estimate the average, and hence, generate the distribution for the averages
- Note: bootstrap method produces the same mean and standard deviation (since the data remains the same)
- However, confidence intervals for a given confidence level: do make sense!
- Bootstrap data does not extend beyond the minimum and maximum values of the given data; confidence intervals obtained from the tails of the distribution could be error prone

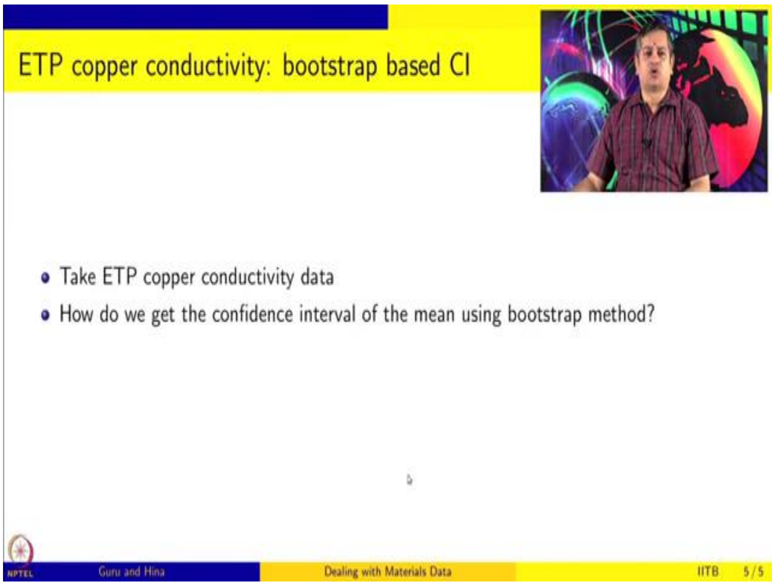
NPTEL Guru and Hina Dealing with Materials Data IITB 4 / 5

The idea in bootstrap method is as follows, you consider the given data. There are n independent measurements of equal weight. But we do not know from which distribution this data is sampled. So this is the assumption that we are making. We are assuming that there are n independent measurements and they are all of equal statistical weight, just that we do not know the distribution. To accurately calculate the mean, you need to know the distribution from which the mean comes which means you need several such samples. And bootstrapping method is a method to generate samples from the existing dataset. It does that by sampling the data with replacement.

And so, you sample it thousands of times for example, you will see an example and then from that calculate the distribution for the mean. And so, from each of this dataset basically we are calculating the average and then getting the distribution for the average. Notice that because we are sampling from the same data with replacement, you cannot really get any more information about mean or standard deviation from the data, it is the same data. So, this mean and standard deviation do not change at all. The reason why we are using the bootstrap method is to get the confidence intervals for the given value like what is the probability that the mean will lie in this range.

One also has to be careful because we are using the same dataset, the minimum value and my maximum value are fixed. So, all the datasets that we generate by this random sampling will also have the same minimum value and same maximum value which means if there is any contribution from the tail that comes, you will be missing out on that information.

(Refer Slide Time: 3:25)

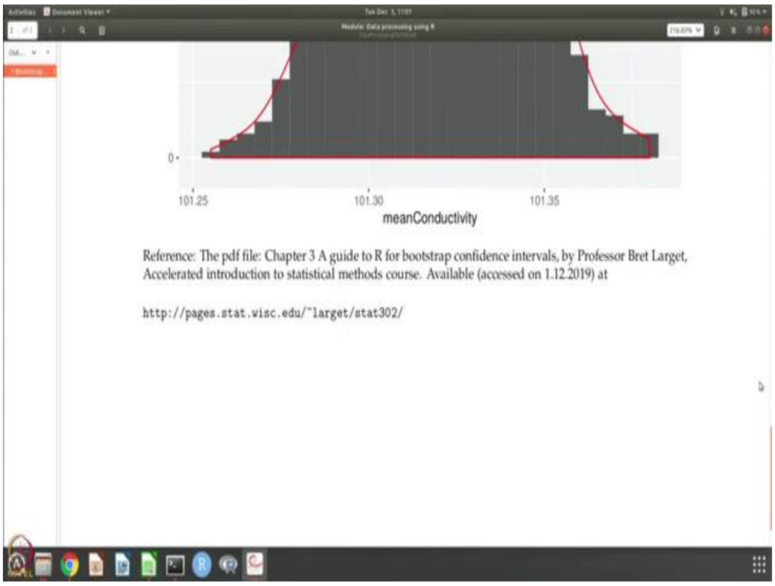


ETP copper conductivity: bootstrap based CI

- Take ETP copper conductivity data
- How do we get the confidence interval of the mean using bootstrap method?

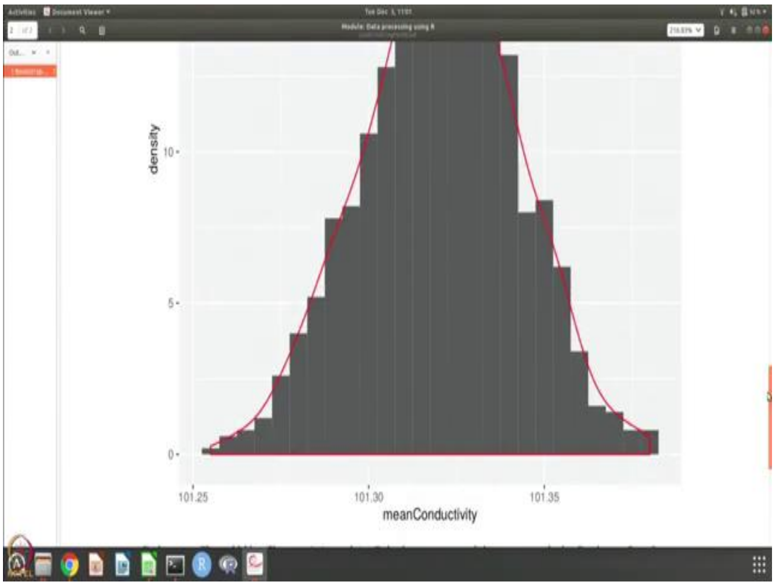
Now, having said that it is still useful to use bootstrap to get confidence intervals, and we will show an example. We will again take the ETP copper conductivity data, we will not assume that it is normally distributed because if you do then we have seen how to get the confidence interval assuming that the standard deviation is known and assuming that the standard deviation is not known, using either normal standard normal or T distribution. But now we just want to do it completely using data. We do not want to assume anything about the underlying distribution. So, can we get the confidence interval? So, for that we have to use the bootstrap method.

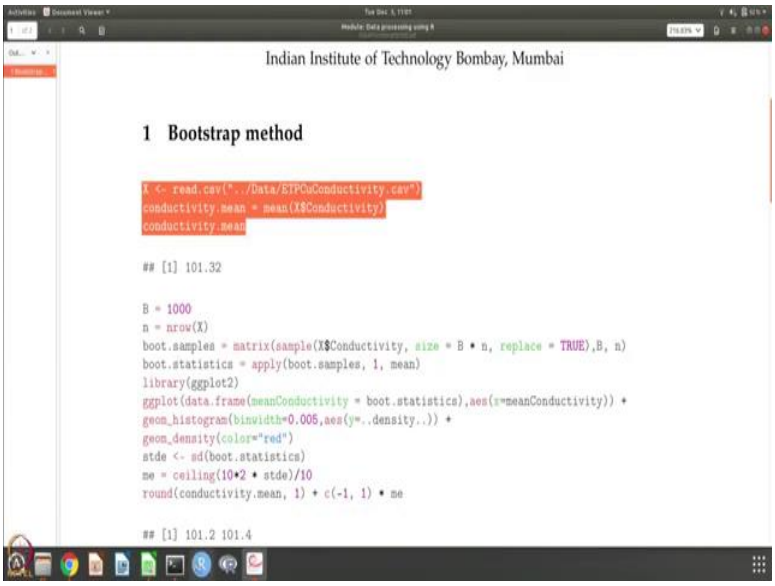
(Refer Slide Time: 4:04)



Reference: The pdf file: Chapter 3 A guide to R for bootstrap confidence intervals, by Professor Bret Larget, Accelerated introduction to statistical methods course. Available (accessed on 1.12.2019) at

http://pages.stat.wisc.edu/~larget/stat302/

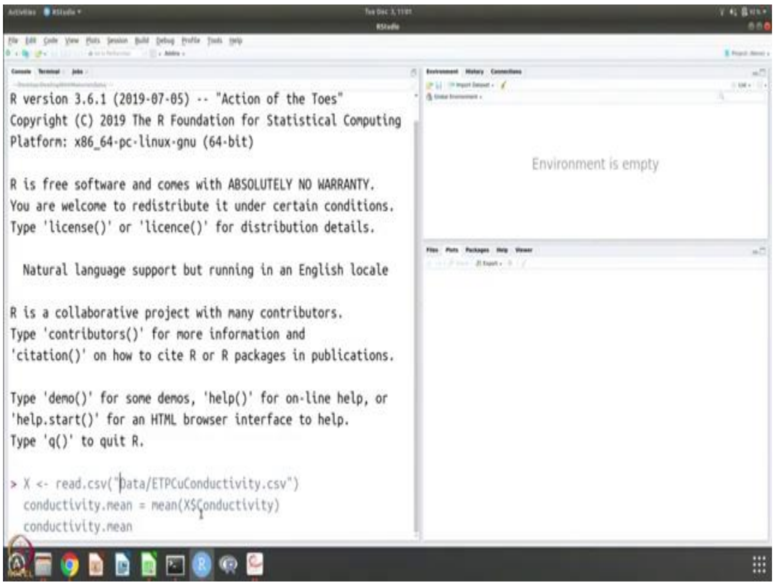




Indian Institute of Technology Bombay, Mumbai

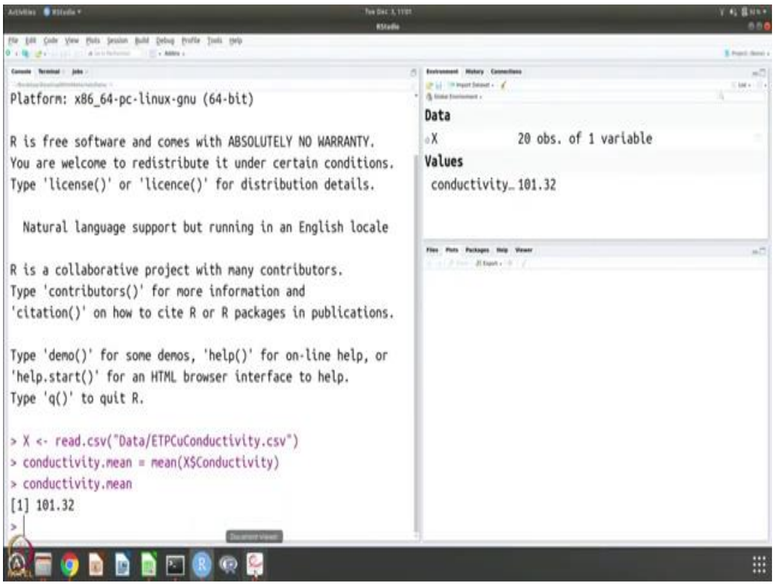
1 Bootstrap method

```
X <- read.csv("../Data/ETPCuConductivity.csv")
conductivity.mean = mean(X$Conductivity)
conductivity.mean

## [1] 101.32

B = 1000
n = nrow(X)
boot.samples = matrix(sample(X$Conductivity, size = B * n, replace = TRUE),B, n)
boot.statistics = apply(boot.samples, 1, mean)
library(ggplot2)
ggplot(data.frame(meanConductivity = boot.statistics),aes(x=meanConductivity)) +
geom_histogram(binwidth=0.005,aes(y=..density..)) +
geom_density(color="red")
stde <- sd(boot.statistics)
me = ceiling(10*2 * stde)/10
round(conductivity.mean, 1) + c(-1, 1) * me

## [1] 101.2 101.4
```



```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> X <- read.csv("Data/ETPCuConductivity.csv")
  conductivity.mean = mean(X$Conductivity)
  conductivity.mean
```

Environment is empty

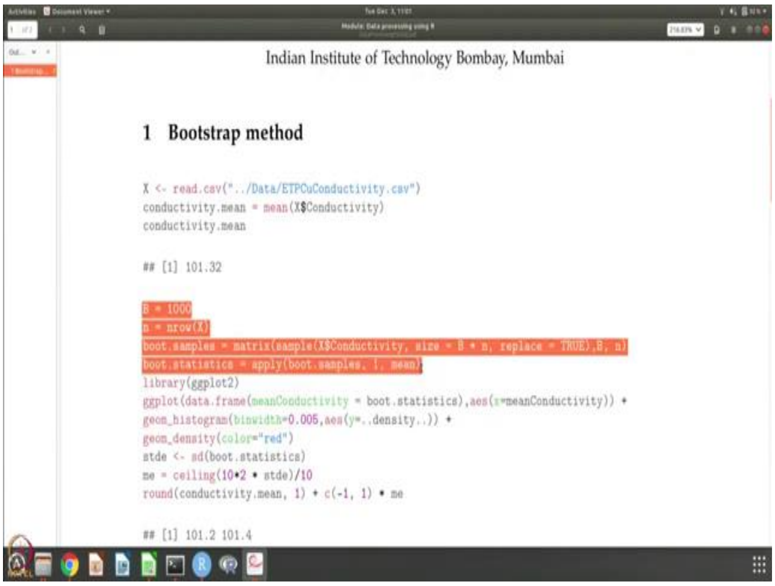


```
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> X <- read.csv("Data/ETPCuConductivity.csv")
> conductivity.mean = mean(X$Conductivity)
> conductivity.mean
[1] 101.32
>
```

Data
X 20 obs. of 1 variable
Values
conductivity… 101.32

Let us do that and for doing that, I also recommend that you go through this chapter 3, a guide to our for bootstrap confidence intervals by Professor Bret Larget and it is part of a statistical course and it is available. So, please do take a look at it and this tutorial is basically based on that chapter. So, let us first take our data and calculate the mean. So that is a first step. So, we read the data and we calculate the mean conductivity. So, that is 101.32. Now, we are very familiar with this dataset. So, we know that that is the number.

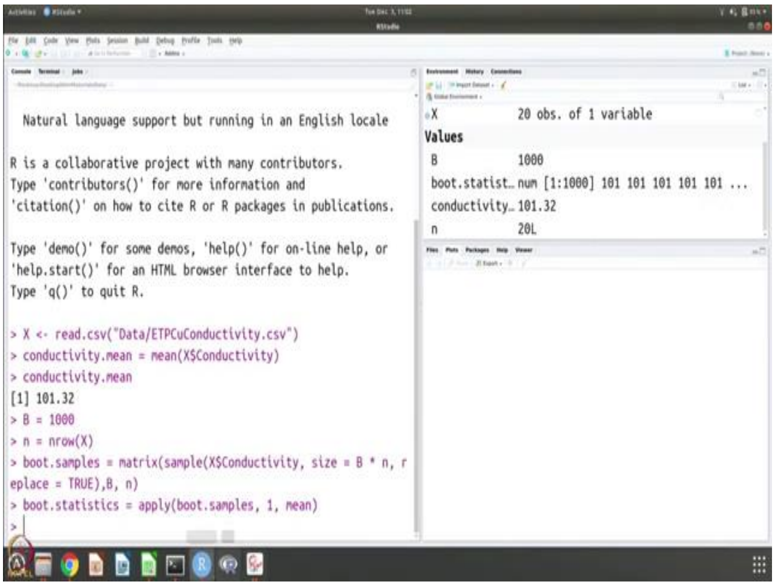
(Refer Slide Time: 4:47)



Indian Institute of Technology Bombay, Mumbai

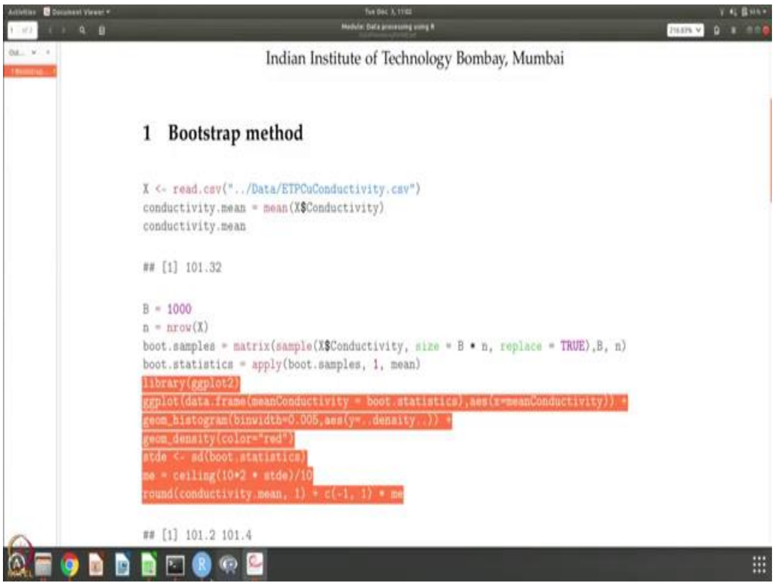
1 Bootstrap method

```
X <- read.csv("../Data/ETPCuConductivity.csv")
conductivity.mean = mean(X$Conductivity)
conductivity.mean

## [1] 101.32

B = 1000
n = nrow(X)
boot.samples = matrix(sample(X$Conductivity, size = B * n, replace = TRUE),B, n)
boot.statistics = apply(boot.samples, 1, mean)
library(ggplot2)
ggplot(data.frame(meanConductivity = boot.statistics),aes(x=meanConductivity)) +
geom_histogram(binwidth=0.005,aes(y=..density..)) +
geom_density(color="red")
stde <- sd(boot.statistics)
me = ceiling(10*2 * stde)/10
round(conductivity.mean, 1) + c(-1, 1) * me

## [1] 101.2 101.4
```



```
 Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> X <- read.csv("Data/ETPCuConductivity.csv")
> conductivity.mean = mean(X$Conductivity)
> conductivity.mean
[1] 101.32
> B = 1000
> n = nrow(X)
> boot.samples = matrix(sample(X$Conductivity, size = B * n, r
eplace = TRUE),B, n)
> boot.statistics = apply(boot.samples, 1, mean)
>
```

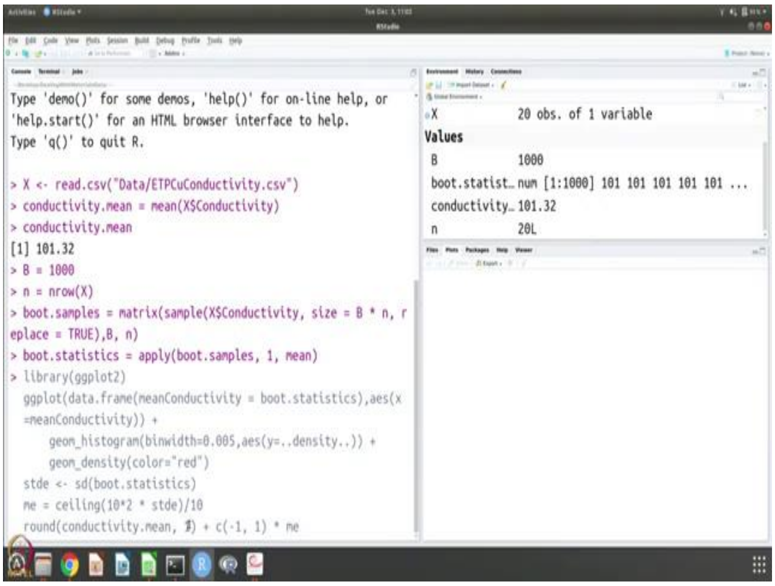
```
X               20 obs. of 1 variable
Values
 B              1000
 boot.statist… num [1:1000] 101 101 101 101 101 ...
 conductivity… 101.32
 n              20L
```



Indian Institute of Technology Bombay, Mumbai

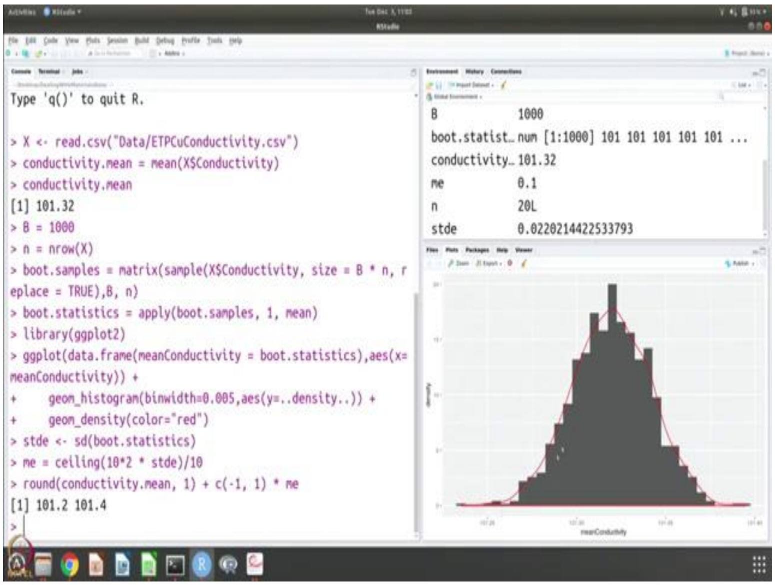
1 Bootstrap method

```
X <- read.csv("../Data/ETPCuConductivity.csv")
conductivity.mean = mean(X$Conductivity)
conductivity.mean

## [1] 101.32

B = 1000
n = nrow(X)
boot.samples = matrix(sample(X$Conductivity, size = B * n, replace = TRUE),B, n)
boot.statistics = apply(boot.samples, 1, mean)
library(ggplot2)
ggplot(data.frame(meanConductivity = boot.statistics),aes(x=meanConductivity)) +
geom_histogram(binwidth=0.005,aes(y=..density..)) +
geom_density(color="red")
stde <- sd(boot.statistics)
me = ceiling(10*2 * stde)/10
round(conductivity.mean, 1) + c(-1, 1) * me

## [1] 101.2 101.4
```



```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> X <- read.csv("Data/ETPCuConductivity.csv")
> conductivity.mean = mean(X$Conductivity)
> conductivity.mean
[1] 101.32
> B = 1000
> n = nrow(X)
> boot.samples = matrix(sample(X$Conductivity, size = B * n, r
eplace = TRUE),B, n)
> boot.statistics = apply(boot.samples, 1, mean)
> library(ggplot2)
  ggplot(data.frame(meanConductivity = boot.statistics),aes(x
  =meanConductivity)) +
      geom_histogram(binwidth=0.005,aes(y=..density..)) +
      geom_density(color="red")
  stde <- sd(boot.statistics)
  me = ceiling(10*2 * stde)/10
  round(conductivity.mean, 1) + c(-1, 1) * me
```

```
X               20 obs. of 1 variable
Values
B               1000
boot.statist…   num [1:1000] 101 101 101 101 101 ...
conductivity…   101.32
n               20L
```



```
Type 'q()' to quit R.

> X <- read.csv("Data/ETPCuConductivity.csv")
> conductivity.mean = mean(X$Conductivity)
> conductivity.mean
[1] 101.32
> B = 1000
> n = nrow(X)
> boot.samples = matrix(sample(X$Conductivity, size = B * n, r
eplace = TRUE),B, n)
> boot.statistics = apply(boot.samples, 1, mean)
> library(ggplot2)
> ggplot(data.frame(meanConductivity = boot.statistics),aes(x=
meanConductivity)) +
+     geom_histogram(binwidth=0.005,aes(y=..density..)) +
+     geom_density(color="red")
> stde <- sd(boot.statistics)
> me = ceiling(10*2 * stde)/10
> round(conductivity.mean, 1) + c(-1, 1) * me
[1] 101.2 101.4
>
```

```
B               1000
boot.statist…   num [1:1000] 101 101 101 101 101 ...
conductivity…   101.32
me              0.1
n               20L
stde            0.0220214422533793
```

So, then what we are going to do? We are going to generate 1000 bootstrap datasets. So, let us do this. So, what does it do? So, we are going to generate 1000 datasets and those are the boot samples. From the sample that the data that we have and in each one, we are going to have the same number of data points, 20 data points. And it is done by sampling with replacement. And then we are going to calculate the statistics for the datasets that we have generated 1000 datasets that we have generated.

So, once we have done that, of course, we want to analyze what is it that we have got and for that we are going to use this. So, what is this? Let us take a look at it. So, first we are going to plot the statistics from the bootstrap data that we have got. And we are going to calculate the error and using the error, we are and this strange formula here is to make sure that we have the error with the right significant digits. And then we are going to say the confidence interval with so much probability that the data will lie in this range. So, that is the, so you can see that the data generated by bootstrapping and the distribution that the, that histogram follows.

And you can also see that the confidence interval is 101.2 to 101.4, which is the same as what you got from the t and normal. So, this is the confidence interval within which the mean will lie. So, to summarize, in addition to knowing the distribution from which the data comes on, so you can estimate the interval, there are distribution free methods as part of robust methods you can use to get an idea about the confidence interval. So this brings us to the end of this module on data processing. So, we will summarize it in the next session. Thank you.