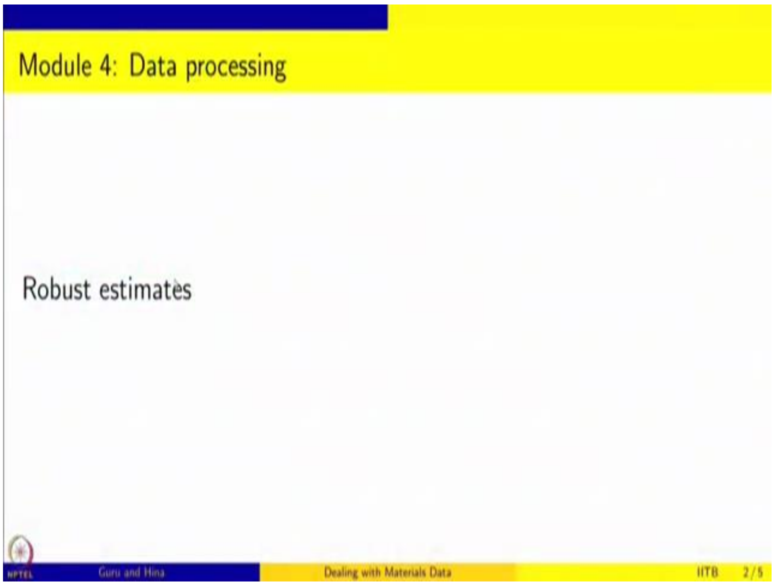
Dealing with Materials Data: Collection, Analysis and Interpretation
Professor M P Gururajan
Department of Metallurgical Engineering and Materials Science
Indian Institute of Technology, Bombay
Lecture 64 - Robust Estimates

Welcome to Dealing with Materials Data, we are looking at Collection, Analysis and Interpretation of Data.

(Refer Slide Time: 0:25)



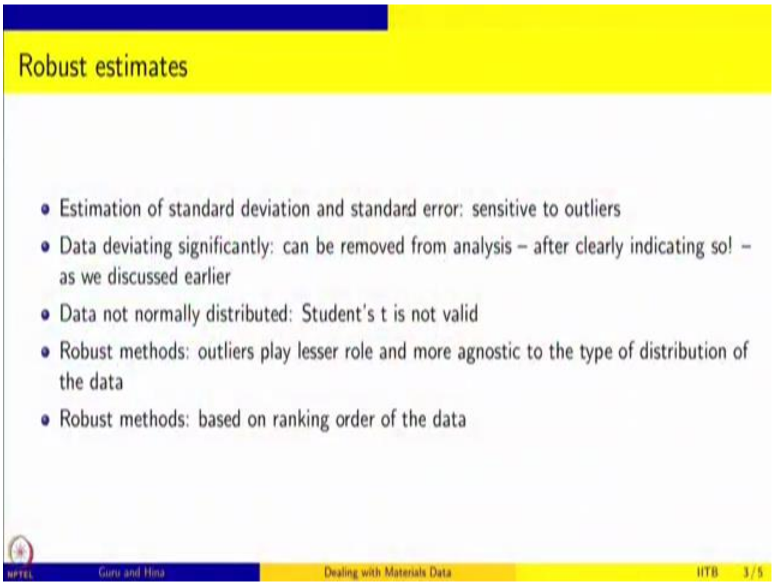
Module 4: Data processing

Robust estimates

NPTEL Guru and Hina Dealing with Materials Data IITB 2/5

We are in the module on data processing and in this session, we will discuss robust estimates.

(Refer Slide Time: 0:33)



Robust estimates

- Estimation of standard deviation and standard error: sensitive to outliers
- Data deviating significantly: can be removed from analysis – after clearly indicating so! – as we discussed earlier
- Data not normally distributed: Student's t is not valid
- Robust methods: outliers play lesser role and more agnostic to the type of distribution of the data
- Robust methods: based on ranking order of the data

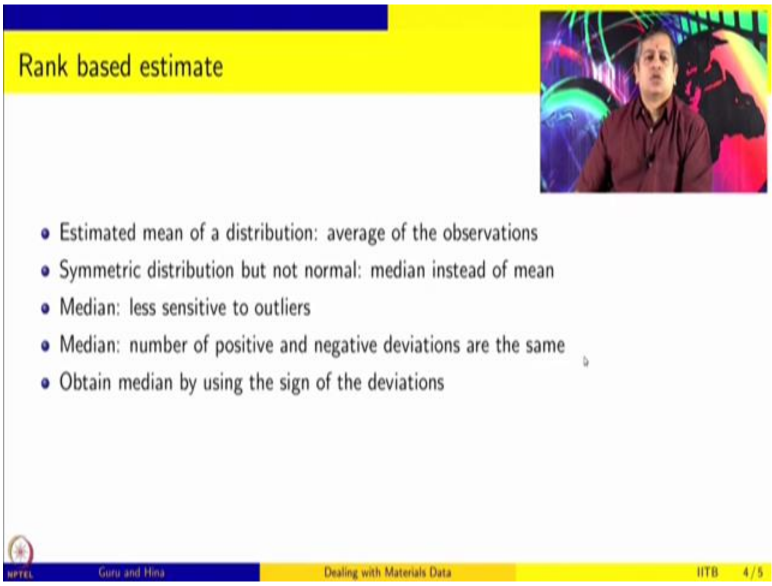
NPTEL Guru and Hina Dealing with Materials Data IITB 3/5

The estimation of standard deviation and standard error that we have been discussing so far is sensitive to outliers. It is because we are calculating the mean squared deviation and if some data point is really far away from the mean, that is going to dominate the analysis. One way to avoid this problem is to remove from analysis the data that is deviating significantly. We have discussed what it means to say some data deviates significantly.

So, if it is more than 3 sigma, 3.5 sigma for example but you should always clearly indicate that you are removing such data. And we have discussed this earlier. And it is always a better idea not to remove data from analysis. So it is better if we can actually come up with an estimation method where we do not have to remove outliers for example. And second problem is that, if the data is not normally distributed, the assumption that we made about normal and student's t, chi squared etcetera are not valid.

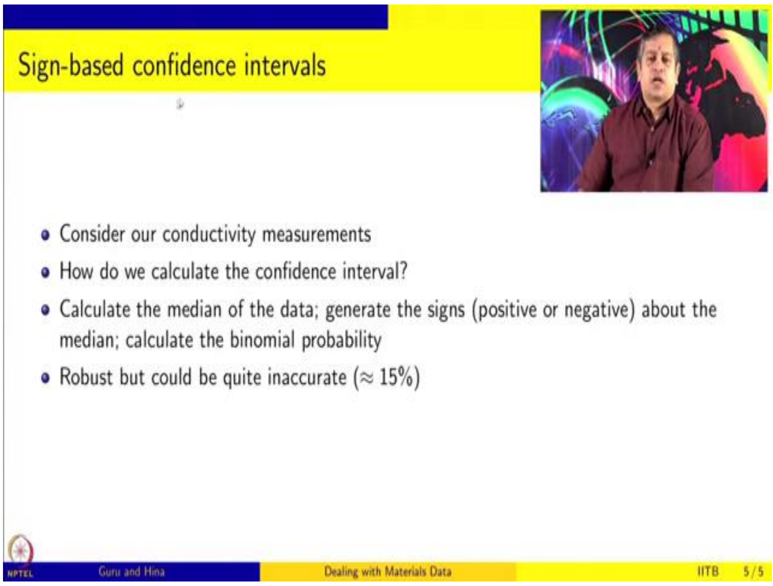
So we want to have estimates for means and spread of data etcetera, which are sort of agnostic to the type of distribution and with outliers playing very nominal role or very less role in determining the quantities. Robust methods are based on ranking order of the data and they are actually methods which can be used to make sure that outliers do not play a significant role and it does not really matter what the type of distribution is.

(Refer Slide Time: 2:20)



Rank based estimate

- Estimated mean of a distribution: average of the observations
- Symmetric distribution but not normal: median instead of mean
- Median: less sensitive to outliers
- Median: number of positive and negative deviations are the same
- Obtain median by using the sign of the deviations



Sign-based confidence intervals

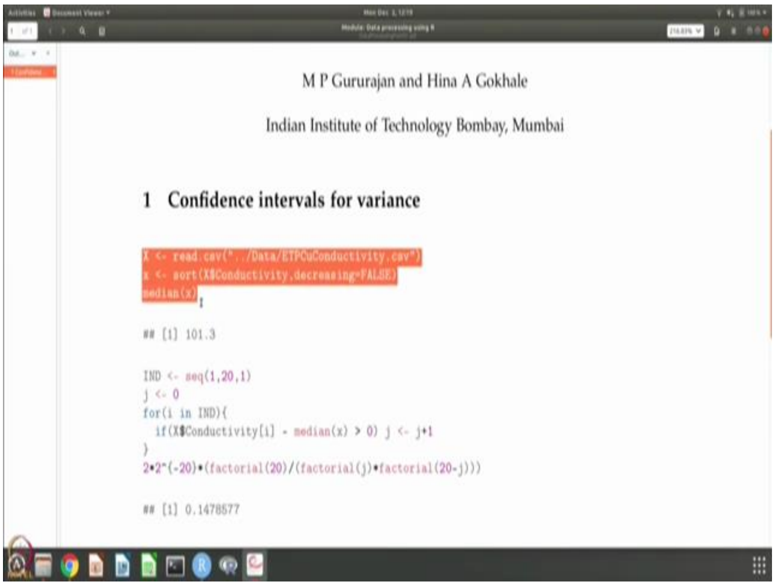
- Consider our conductivity measurements
- How do we calculate the confidence interval?
- Calculate the median of the data; generate the signs (positive or negative) about the median; calculate the binomial probability
- Robust but could be quite inaccurate ($\approx 15\%$)

So one estimate, so rank based estimate, for example. The estimated mean of a distribution is basically average of the observations. Suppose, if you assume that your distribution is symmetric, but it is not normal, you can use median instead of mean, because mean is sensitive to outliers but median is not. So, you can use median. And median also has this property that about median if you go on either side, the number of positive and negative deviations are the same.

So, you can obtain median by using the sign of the deviations for example. Now you can also give confidence intervals which are sign based. Let us consider our conductivity measurement as an example. So how do we calculate confidence intervals for this case, which is based on the sign based confidence interval? We calculate the median and we generate the signs positive or negative about this median.

And we calculate the binomial probability for so many positives and so many negatives or successes and failures, right, pluses and minuses. It could be a robust method, but it could be quite inaccurate also. So, we will do this analysis and see how it works out for our conductivity data.

(Refer Slide Time: 4:05)

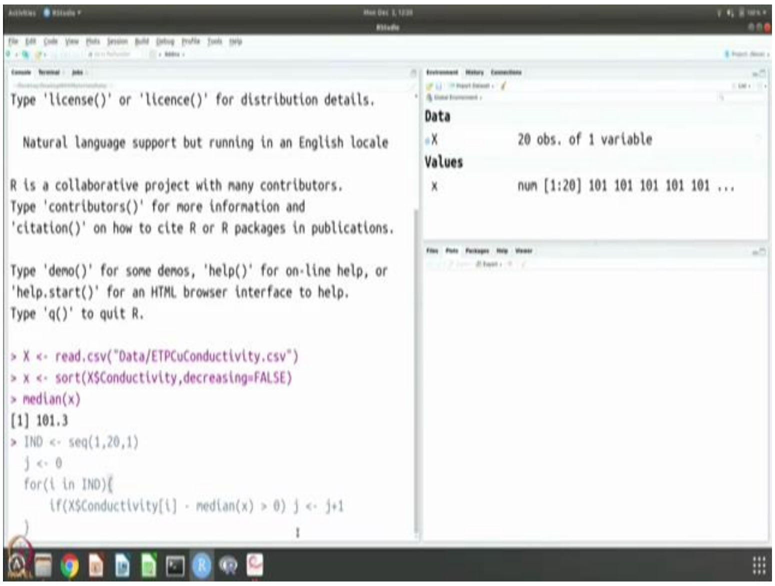


M P Gururajan and Hina A Gokhale

Indian Institute of Technology Bombay, Mumbai

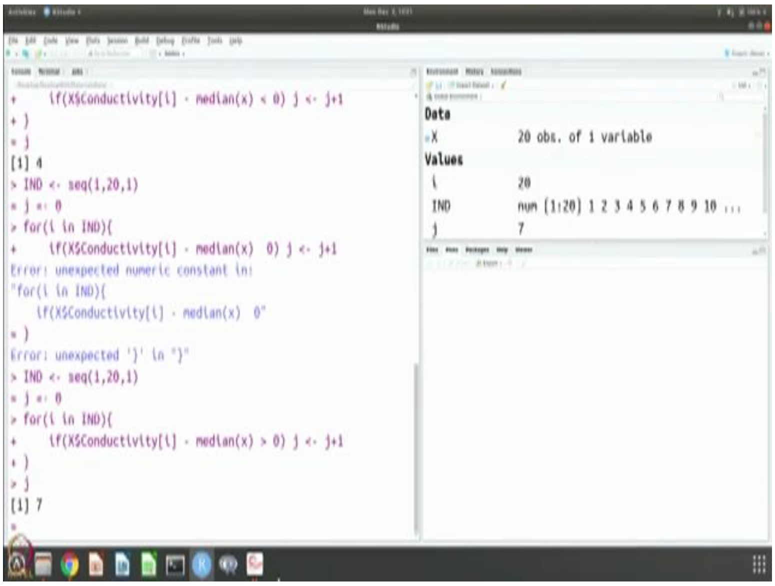
1 Confidence intervals for variance

```
X <- read.csv("../Data/ETPCuConductivity.csv")
x <- sort(X$Conductivity,decreasing=FALSE)
median(x)

## [1] 101.3

IND <- seq(1,20,1)
j <- 0
for(i in IND){
  if(X$Conductivity[i] - median(x) > 0) j <- j+1
}
2*2^(-20)*(factorial(20)/(factorial(j)*factorial(20-j)))

## [1] 0.1478577
```



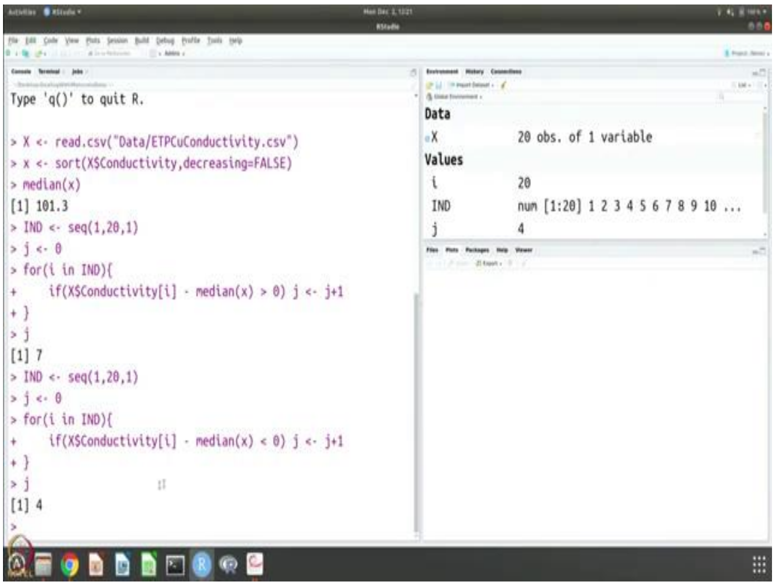
```
Type 'license()' or 'licence()' for distribution details.

  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")
> x <- sort(X$Conductivity,decreasing=FALSE)
> median(x)
[1] 101.3
> IND <- seq(1,20,1)
  j <- 0
  for(i in IND){
      if(X$Conductivity[i] - median(x) > 0) j <- j+1
  }
```

```
Data
X             20 obs. of 1 variable
Values
x             num [1:20] 101 101 101 101 101 ...
```



```
+     if(X$Conductivity[i] - median(x) < 0) j <- j+1
+ }
> j
[1] 4
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x)  0) j <- j+1
Error: unexpected numeric constant in:
"for(i in IND){
    if(X$Conductivity[i] - median(x)  0"
> }
Error: unexpected '}' in "}"
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x) > 0) j <- j+1
+ }
> j
[1] 7
>
```

```
Data
X             20 obs. of 1 variable
Values
i             20
IND           num [1:20] 1 2 3 4 5 6 7 8 9 10 ...
j             7
```



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

> X <- read.csv("Data/ETPCuConductivity.csv")
> x <- sort(X$Conductivity,decreasing=FALSE)
> median(x)
[1] 101.3
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x) > 0) j <- j+1
+ }
> j
[1] 7
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x) < 0) j <- j+1
+ }
> j
[1] 4
>
```

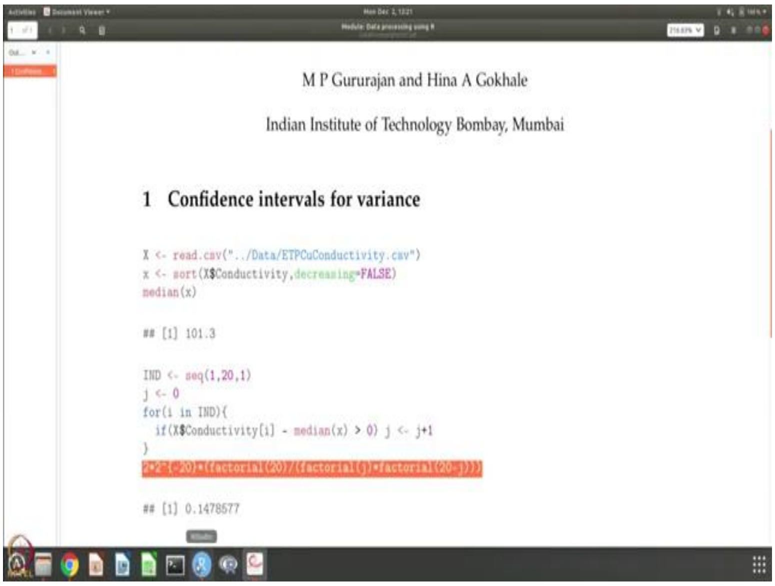
```
Data
X          20 obs. of 1 variable
Values
i          20
IND        num [1:20] 1 2 3 4 5 6 7 8 9 10 ...
j          4
```

So, we have the data, conductivity data, and we are going to get the median. And we have also sorted the data in increasing order and the median is 101.3. So, now what we are going to do, we are going to count…So, what this is doing is that okay, there are 20 data points, so we start this j with the value of 0 and we go and calculate this conductivity value minus median. If it is greater than 0, we say that j increases by 1. So basically it tells you how many data points are greater than median.

And of course, you can also do less than, and that will give the exact opposite. So, let us do this and what is the j value you can see j is 7. And if you did the other way, j should be 13 because that gives you the number of times it is.

So in this case, you do not see because there are conductivity values which are exactly equal to the median value itself. So it is neither greater than nor less than but equal to. So in the case, where it is greater than 0, so we found j is to be 7. So now we calculate the binomial probability distribution for so many.

(Refer Slide Time: 6:21)

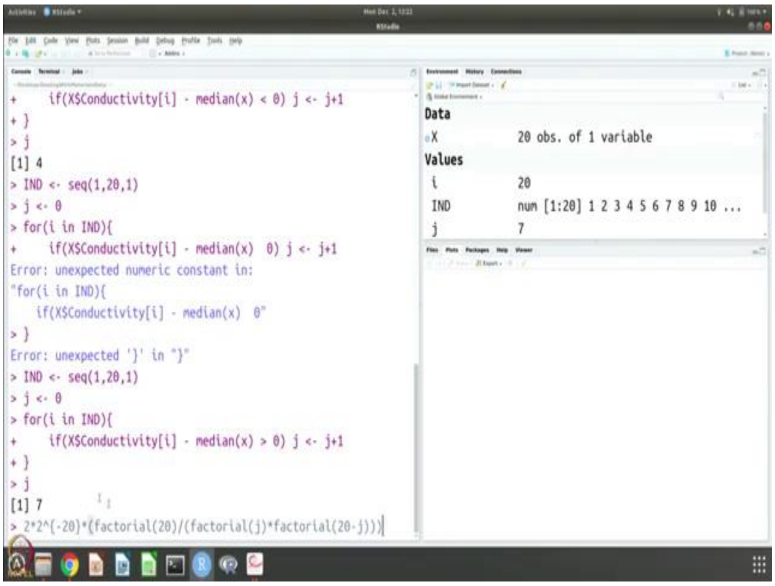


M P Gururajan and Hina A Gokhale

Indian Institute of Technology Bombay, Mumbai

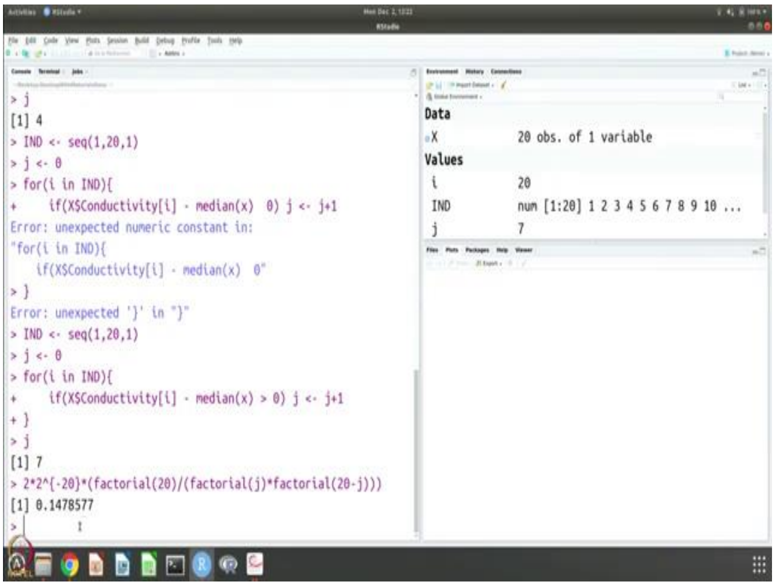
1 Confidence intervals for variance

```
X <- read.csv("../Data/ETPCuConductivity.csv")
x <- sort(X$Conductivity,decreasing=FALSE)
median(x)

## [1] 101.3

IND <- seq(1,20,1)
j <- 0
for(i in IND){
  if(X$Conductivity[i] - median(x) > 0) j <- j+1
}
2*2^(-20)*(factorial(20)/(factorial(j)*factorial(20-j)))

## [1] 0.1478577
```



```
+     if(X$Conductivity[i] - median(x) < 0) j <- j+1
+ }
> j
[1] 4
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x)  0) j <- j+1
Error: unexpected numeric constant in:
"for(i in IND){
    if(X$Conductivity[i] - median(x)  0"
> }
Error: unexpected '}' in "}"
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x) > 0) j <- j+1
+ }
> j
[1] 7
> 2*2^(-20)*(factorial(20)/(factorial(j)*factorial(20-j)))
```

```
Data
X        20 obs. of 1 variable
Values
i        20
IND      num [1:20] 1 2 3 4 5 6 7 8 9 10 ...
j        7
```



```
> j
[1] 4
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x)  0) j <- j+1
Error: unexpected numeric constant in:
"for(i in IND){
    if(X$Conductivity[i] - median(x)  0"
> }
Error: unexpected '}' in "}"
> IND <- seq(1,20,1)
> j <- 0
> for(i in IND){
+     if(X$Conductivity[i] - median(x) > 0) j <- j+1
+ }
> j
[1] 7
> 2*2^(-20)*(factorial(20)/(factorial(j)*factorial(20-j)))
[1] 0.1478577
>
```

```
Data
X        20 obs. of 1 variable
Values
i        20
IND      num [1:20] 1 2 3 4 5 6 7 8 9 10 ...
j        7
```

So this 2 accounts for the fact that it is factorial 20 j, factorial j factorial 20 minus j. You can also have factorial 20 by factorial 20 minus j factorial j. So, that is for 2 and 2 to the power minus 20 because for each case it is either plus or minus. So, that is 0.5 probability and there are 20 of them. So, 1 by 2 to the power 20 is what this value is. So, this gives you something like 0.15. So, in other words, the probability that the median lies in this range is about 15 percent.

As you can see, it is not very accurate, there are better estimates and you can get better numbers from those estimates. But, this is a very robust estimate because it does not assume anything about the underlying probability distribution. And so, such rank based estimates in many cases are also very useful. So, we will continue, there is one more robust estimate which is based on bootstrap method.

And that will bring us to the end of this module on simple data processing. We will continue with more involved data processing in terms of regression and fitting and so on. But before we do that, we will look at bootstrap method as the last method for this kind of estimation. Thank you.