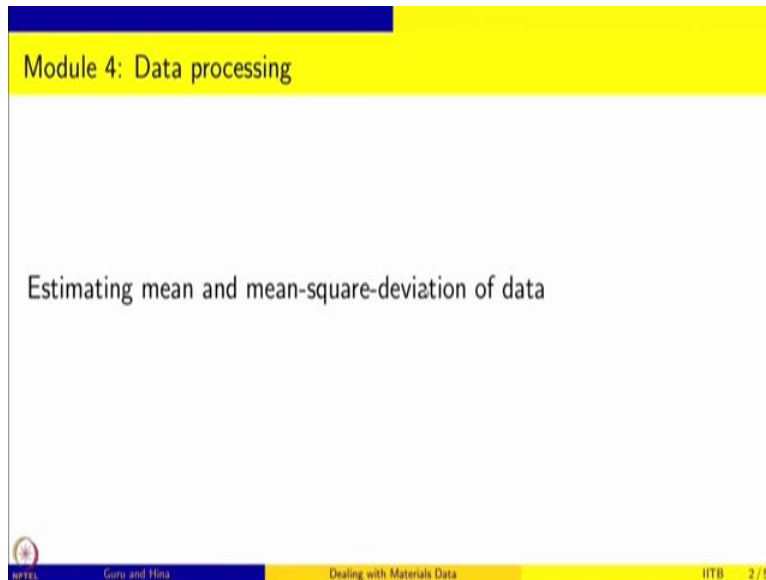


Dealing with Materials Data
Professor M P Gururajan
Professor Hina A Gokhale
Department of Metallurgical Engineering and Materials Science
Indian Institute of Technology, Bombay
Lecture 62
Estimating Mean and Mean-Square-Deviation of Data


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: 0:26)



Module 4: Data processing

Estimating mean and mean-square-deviation of data

 Guru and Hina Dealing with Materials Data IITB 2/5

Estimating properties of the data

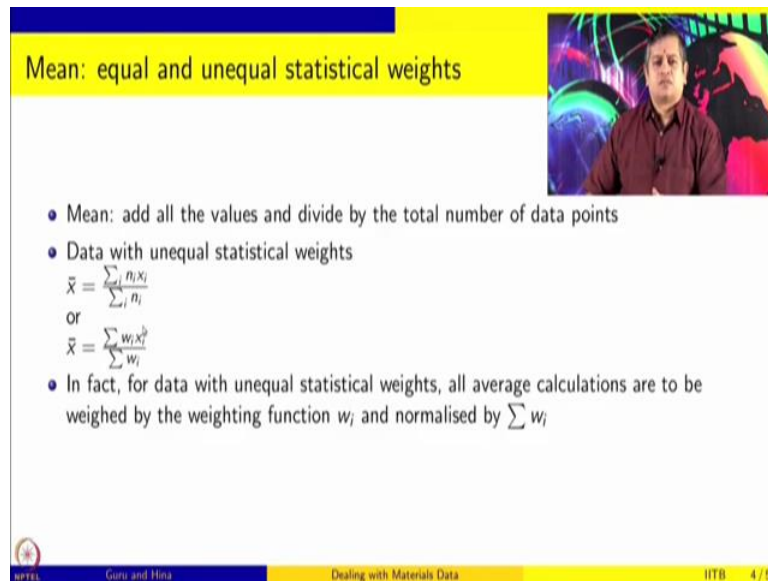
- Mean of a data
- Mean-squared deviation (M.S.D)
- Root-mean-squared deviation (R.M.S.D)



And in this session we are going to learn about estimating the mean and the mean squared deviation of data. So, these are the properties of data, the mean of the data basically tells there the, if you assume that the data is normally distributed, for example, it tells you actually what is the value about which you see a spread.

But if you do not assume anything about the distribution of the data, the mean also tells you the likelihood estimate or maximum likelihood estimate for the given parameter. Mean-squared deviation basically tells you the spread of the data and root mean-squared deviation is just a root of this quantity. So, it is also a measure of the spread of the data.

(Refer Slide Time: 01:22)



Mean: equal and unequal statistical weights

- Mean: add all the values and divide by the total number of data points
- Data with unequal statistical weights
$$\bar{x} = \frac{\sum_i n_i x_i}{\sum_i n_i}$$
or
$$\bar{x} = \frac{\sum_i w_i x_i}{\sum_i w_i}$$
- In fact, for data with unequal statistical weights, all average calculations are to be weighed by the weighting function w_i and normalised by $\sum w_i$

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

So, mean is very simple. So, you add all values and divide by the total number of data points. So, if you have data with unequal statistical weights, then you have to make sure that you weigh by the weight and then take the average.

$$\bar{x} = \frac{\sum_i n_i x_i}{\sum_i n_i}$$

or

$$\bar{x} = \frac{\sum w_i x_i}{\sum w_i}$$

And if you take the, these w_i themselves normalized in such a way that summation w_i is 1, it is just summation $w_i x_i$. We will see an example, we will look at the cluster size frequency data that we have digitized and used to look at taking means of this type.

(Refer Slide Time: 02:21)

M.S.D and R.M.S.D from average

- Mean-Squared-Deviation from the average: $\frac{1}{n}(x_i - \text{mean})^2$
- Root-Mean-Squared-Deviation from the average: $\sqrt{\text{M.S.D}}$

MPYU Guru and Hina Dealing with Materials Data IITB 5/5

Mean squared deviation from average is nothing but, for every value you subtract the mean you square, so it is the mean squared deviation. It is this mean because you then take the mean of those values.

$$\text{Mean-squared-Deviation from the average: } \frac{1}{n}(x_i - \text{mean})^2$$

And root mean squared deviation from the average is just the square root of this value, this is MSD and so RMSD is just square root of this. So let us do this for the data that we have.

(Refer Slide Time: 02:58)

Module: Data processing using R

M P Gururajan and Hina A Gokhale

Indian Institute of Technology Bombay, Mumbai

1 Averages

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

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.

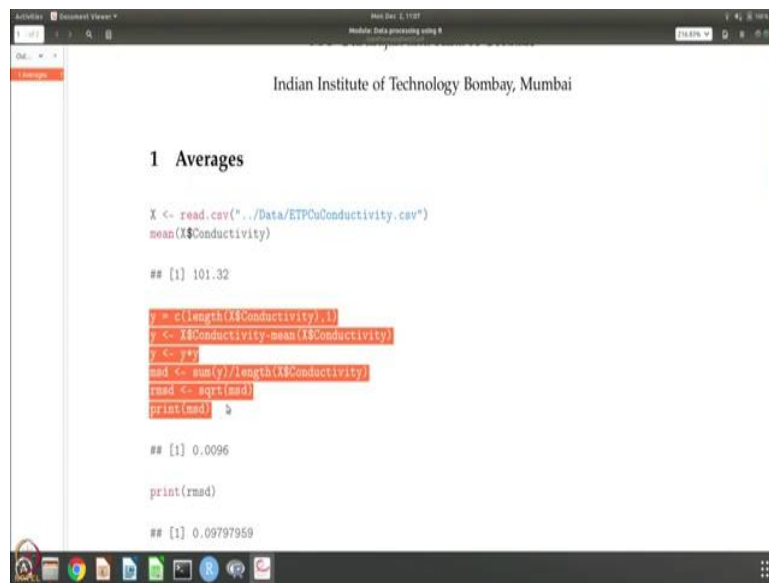
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

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

Data
X 20 obs. of 1 variable

Let us start R and let us start with this exercise. So I am going to do the DP connectivity data and mean is of course, 101.32, that is straightforward.

(Refer Slide Time: 3:24)



```
Indian Institute of Technology Bombay, Mumbai

1 Averages

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

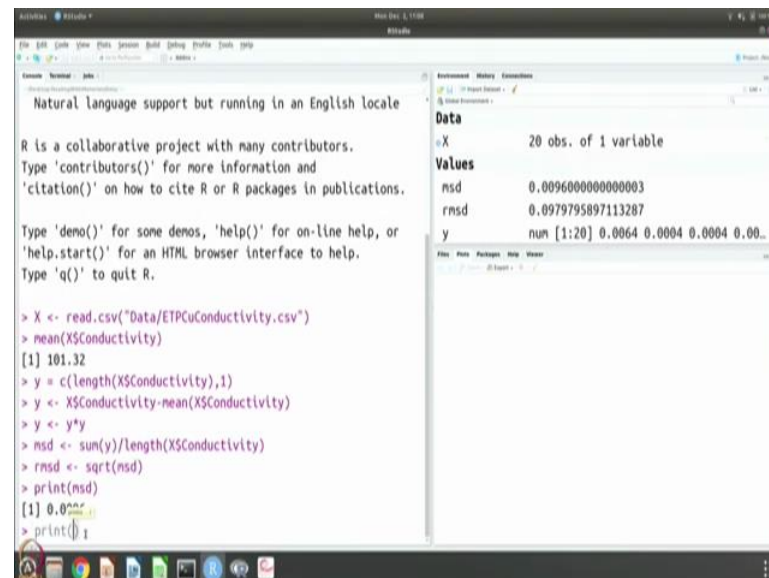
## [1] 101.32

y <- c(length(X$Conductivity),1)
y <- X$Conductivity-mean(X$Conductivity)
y <- y*y
msd <- sum(y)/length(X$Conductivity)
rmsd <- sqrt(msd)
print(msd)

## [1] 0.0096

print(rmsd)

## [1] 0.09797959
```



```
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.

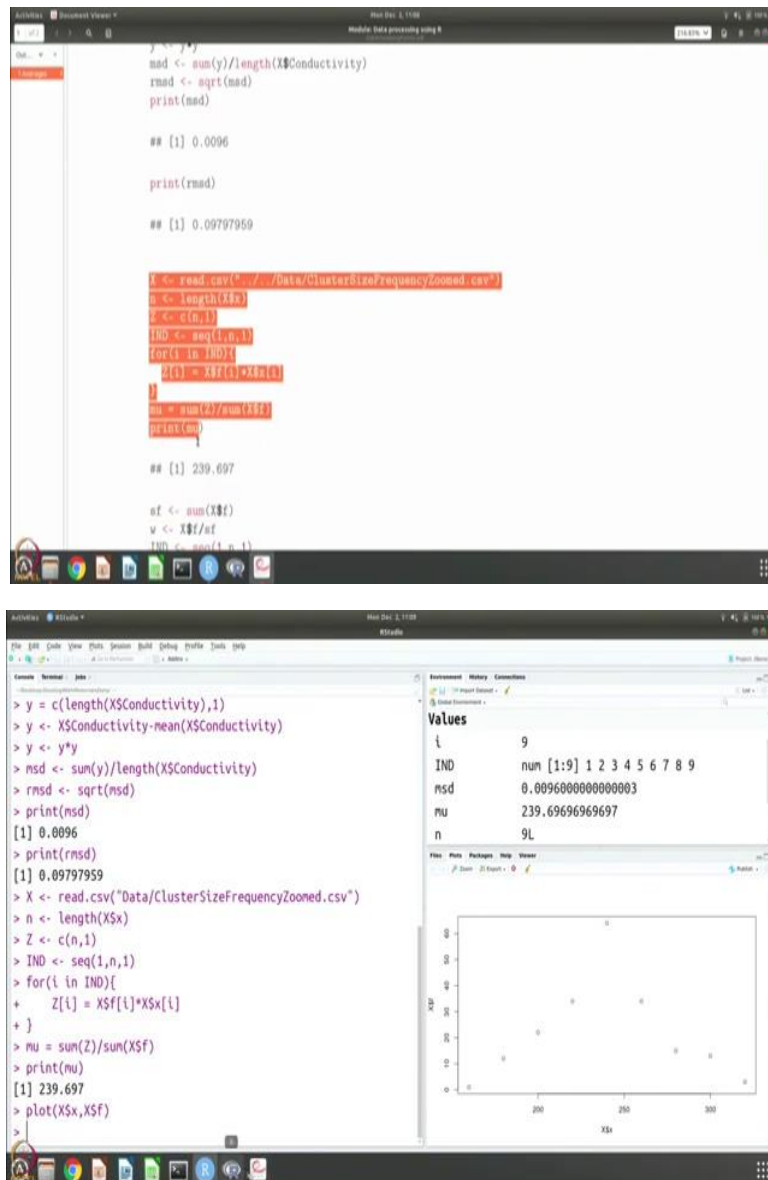
> X <- read.csv("Data/ETPCuConductivity.csv")
> mean(X$Conductivity)
[1] 101.32
> y = c(length(X$Conductivity),1)
> y <- X$Conductivity-mean(X$Conductivity)
> y <- y*y
> msd <- sum(y)/length(X$Conductivity)
> rmsd <- sqrt(msd)
> print(msd)
[1] 0.0096
> print(rmsd)
[1] 0.09797959
```

Data	
eX	20 obs. of 1 variable
Values	
msd	0.0096000000000003
rmsd	0.0979795897113287
y	num [1:20] 0.0064 0.0004 0.0004 0.0004 0.00...

You can calculate the, okay, so what did we do? We took the conductivity and subtracted the mean from every value, and we squared it, and we took the sum of all the squares and divided by the total number of points, so that was mean squared deviation. And of course, you can also print the root means squared deviation.

So you can see that 0.0096 is the mean squared deviation and the 0.09797959 is the root mean squared deviation.

(Refer Slide Time: 04:14)



And how do we do this for data? Okay, let us do that exercise also with different statistical weights right. So, that is what we want to do. First let us calculate the mean, the average. Okay, so we need to read the data and we need to decide how many data points are there. And then we need to give the weight, the frequency and then we have to divide by the sum of the frequency.

Because this is the w_i , the frequency is the statistical weight. And we are going to add them all up so that we will normalize it. So, this is the way to calculate the average. So this is 239 and because the data as you have seen. So if you plot, you will see that the peak is somewhere around 240. So, the average turns out to be 240, so that is expected.

(Refer Slide Time: 05:34)

```
## [1] 0.09797959

X <- read.csv("../Data/ClusterSizeFrequencyZoomed.csv")
n <- length(X$x)
Z <- c(n,1)
IND <- seq(1,n,1)
for(i in IND){
  Z[i] = X$f[i]*X$x[i]
}
mu = sum(Z)/sum(X$f)
print(mu)

## [1] 239.697

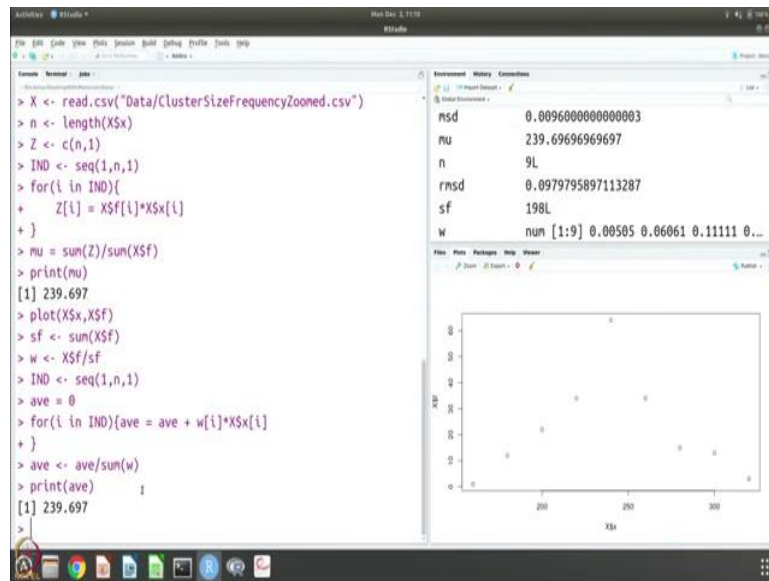
#f <- sum(X$f)
v <- X$f/v
IND <- seq(1,n,1)
ave <- v
for(i in IND){
```

```
1

ave = ave + w[i]*X$x[i]
}
ave <- ave/sum(w)
print(ave)

## [1] 239.697

msd = 0
for(i in IND){
  msd = msd + w[i]*(X$x[i]-ave)*(X$x[i]-ave)
}
msd <- msd/sum(w)
print(msd)
```

Now, let us calculate the mean squared deviation and root mean squared deviation. Okay, so this is another way you can get the weights themselves normalized first, so that they add up to 1 and of course you will get the same number because it is just the algebra, nothing else is different.

(Refer Slide Time: 06:13)

```
ave = ave + w[l]*Xx[l]
}
ave <- ave/sum(w)
print(ave)

## [1] 239.697

msd = 0
for(l in IND){
  msd = msd + w[l]*(Xx[l]-ave)*(Xx[l]-ave)
}
msd <- msd/sum(w)
print(msd)
```

The screenshot shows the RStudio interface. The console on the left contains the following R code and output:

```
[1] 239.697
> plot(Xx,Xsf)
> sf <- sum(Xsf)
> w <- Xsf/sf
> IND <- seq(1,n,1)
> ave = 0
> for(l in IND){ave = ave + w[l]*Xx[l]
+ }
> ave <- ave/sum(w)
> print(ave)
[1] 239.697
> msd = 0
> for(l in IND){
+   msd = msd + w[l]*(Xx[l]-ave)*(Xx[l]-ave)
+ }
> msd <- msd/sum(w)
> print(msd)
[1] 1020.11
> sqrt(msd)
[1] 31.93916
>
```

The Environment pane on the right shows the following variables:

l	9
IND	num [1:9] 1 2 3 4 5 6 7 8 9
msd	1020.11019283747
mu	239.69696969697
n	9L
rmsd	0.0979795897113287

The Plot pane shows a scatter plot of Xx versus Xsf. The x-axis is labeled 'Xx' and ranges from 0 to 300. The y-axis is labeled 'Xsf' and ranges from 0 to 100. There are 9 data points plotted as small squares.

So now let us calculate the mean squared deviation. So how do we calculate the mean squared deviation? You can see that it is the same way. So we take each x value and subtract the average and square the value. But now this has to be weighed by the weighting function. And weighting function is something that we just now calculated.

So we are and remember it is the other type of waiting function, so I took all the frequencies. First I summed all the frequencies, so divided by it. So I have the weighing factor. This weighting factor is what I am going to use. So this is a statistical weight for me to do the calculations. So mean squared deviation is nothing but mean squared deviation and this sum w is going to give 1, so that is the mean squared deviation.

So I see mean squared deviation is 1020 and root mean square deviation of course is nothing but the square root of MSD. So, you get about 32 as these spread of this data which is what is given here okay.

So, to summarize, in this session we have seen that you can take data, it could be raw data or it could be data with unequal statistical weights that you got from some analyzed data. In both cases you can calculate the average and you can calculate the spread of the data by looking at how far away from the average the data points lie. And so, we use mean squared deviation and root mean squared deviation to get this spread of the data.

So, we have shown that for 2 cases, one is copper conductivity, the other one is the particle size of titanium aggregates that we have taken from the literature. So we will have more such exercises during these weeks sessions for you to become familiar with this kind of analysis. Thank you.