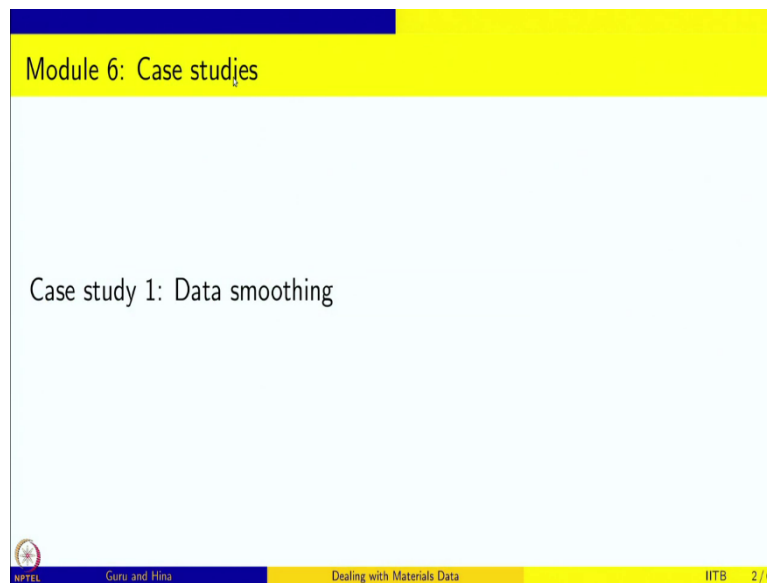


Dealing with Materials Data: Collection, Analysis and Interpretation
Professor M P Gururajan
Professor Hina A Gokhale
Department of Metallurgical Engineering and Materials Science
Indian Institute of Technology, Bombay
Lecture No. 93
Case study 1: Data smoothing 1

Welcome to dealing with materials data. This is a course on the Collection Analysis and Interpretation of Data from Material Science and Engineering.

(Refer Slide Time: 0:25)



We are in module 6 which is in case studies and first case study that we are going to take up is on Data Smoothing.

(Refer Slide Time: 0:43)

Mechanical properties and testing

- Universal Testing Machine
- Impose load and measure displacement
- Stress-strain: a typical experiment to determine mechanical properties of a material
- Elastic modulus (stiffness)
- Strength (Yield strength, Ultimate Tensile Strength, 0.2 percent proof stress)
- Resilience
- Ductility, Toughness
- After test, most machines will give modulus, strength etc



Mechanical properties are very commonly tested and measured because for many materials these are important and universal testing machine – UTM is the machine that is typically used to get some of these mechanical properties. And in these machines we impose a load and measure the displacement and from these we get the stress and strain and a typical experiment is consist of tensile stresses and the corresponding strains.

And then you can determine a whole bunch of mechanical properties, elastic modulus or stiffness of the material, strength which is the yield strength or ultimate tensile strength are 0.2 percent proof stress, so there are lots of measures for knowing strengths of the material. Resilience which is the area under the curve in the elastic region and ductility, which is the strain that you get when the material fails and toughness, which is the area under the stress strain curve.

So all this you can get and in most of the machines after test there will be a computer that is attached to the machine which also collects the data and the data will be analysed and the machine will give you the modulus strength, etc. But in this exercise, just to have better understanding of what goes in in calculating these measures and also to appreciate some of these nuances involved in getting it automatically like that in a machine. We are going to do this by ourselves. We are going to do this by hand.

(Refer Slide Time: 2:12)

Activities LibreOffice Calc Thu Dec 12, 10:31
Al_tensiledata.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

LibreOffice Calc 10

Average: 3.38 Sum: 3.38

	A	B	C	D	E	F	G	H
52	2.19	0						
53	2.15	0						
54	2.14	0						
55	2.23	0						
56	2.33	0						
57	2.36	0						
58	2.55	0						
59	2.59	0						
60	2.53	0						
61	2.72	0						
62	2.89	0						
63	3.13	0						
64	3.37	0						
65	3.38	0.01						
66	3.37	0.01						
67	3.46	0.01						

Al_tensiledata Default English (India) Average: 3.38 Sum: 3.38 200%

Activities LibreOffice Calc Thu Dec 12, 10:31
Al_tensiledata.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

LibreOffice Calc 10

Average: 24.99 Sum: 24.99

	A	B	C	D	E	F	G	H
195	18.4	0.04						
196	19.04	0.04						
197	19.58	0.04						
198	19.64	0.04						
199	19.61	0.04						
200	19.98	0.04						
201	20.7	0.04						
202	21.39	0.05						
203	21.87	0.05						
204	22.06	0.05						
205	21.96	0.05						
206	22.12	0.05						
207	23.18	0.05						
208	24.18	0.05						
209	24.28	0.05						
210	24.89	0.05						

Al_tensiledata Default English (India) Average: 24.99 Sum: 24.99 200%

Activities LibreOffice Calc Thu Dec 12, 10:32
Al_tensiledata.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

LibreOffice Calc 10

Average: 30.78 Sum: 30.78

	A	B	C	D	E	F	G	H
218	28.13	0.05						
219	28.79	0.05						
220	29.66	0.06						
221	30.56	0.06						
222	30.46	0.06						
223	30.45	0.06						
224	30.78	0.06						
225	32.13	0.06						
226	33.01	0.06						
227	33.75	0.06						
228	34.13	0.06						
229	34.41	0.06						
230	34.53	0.06						
231	35.92	0.06						
232	36.7	0.07						
233	37.93	0.07						

Al_tensiledata Default English (India) Average: 30.78 Sum: 30.78 200%

Stress	Strain
40.07	0.07
40.93	0.07
42.24	0.07
42.26	0.07
42.27	0.07
42.54	0.07
44.15	0.07
45.04	0.08
46.35	0.08
46.24	0.08
46.22	0.08
46.8	0.08
48.17	0.08
48.95	0.08
49.96	0.08
50.39	0.09

And I want to show a typical data that you will see for a stress strain. For example, here you see that this is for aluminium and this is stress and strain and you see that even though it is a tensile test, stress shown as negative value and what is more? There is some stress but the strain is still shown to be 0, right. So for, at the beginning of the test before things settle down and so here is the first time you will see a non-zero strain.

On the other hand, when you have stress to be non-zero you would expect the strain to be non-zero but at least the machine is not able to measure these quantities or it thinks it is zero or it is measuring it as wrongly as zero. So in any case, so this is the kind of data we have and on top of it after this also, there are, so you have several stress values and the strain remains constant, maybe because it is not able to distinguish between the different strains.

For example, 30.56, in fact, 39.66, and 35.92, they all show 0.6, 0.06. But probably there is some change in the third decimal place the machine is not able to measure that and so on and so forth, and you will also see when we plot that the data is a little bit noisy, so we will do this plotting and see.

(Refer Slide Time: 3:59)

Stress-strain curve

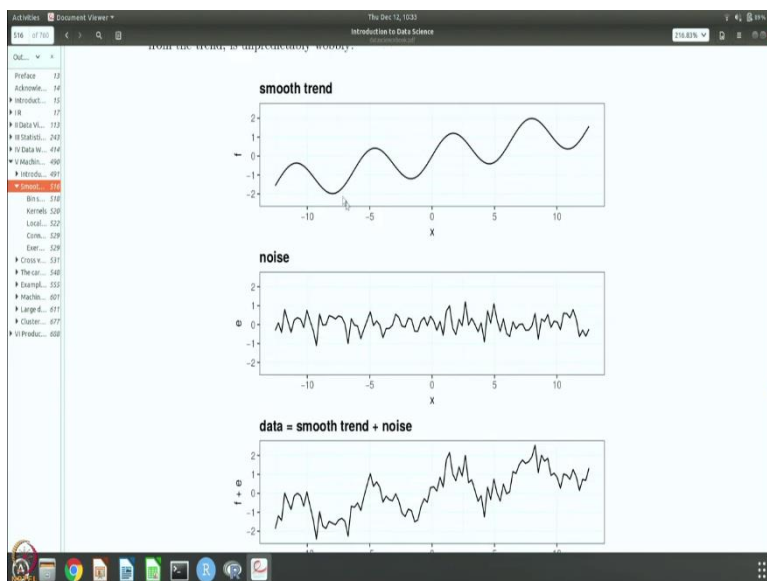
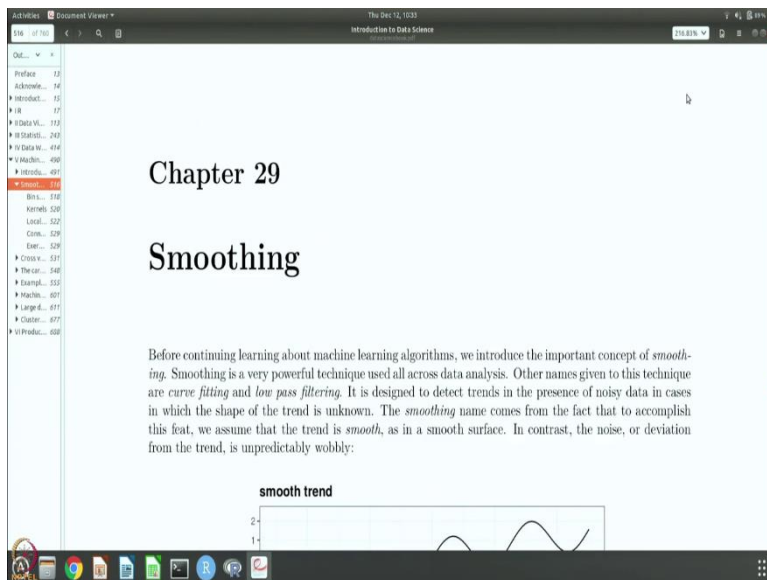
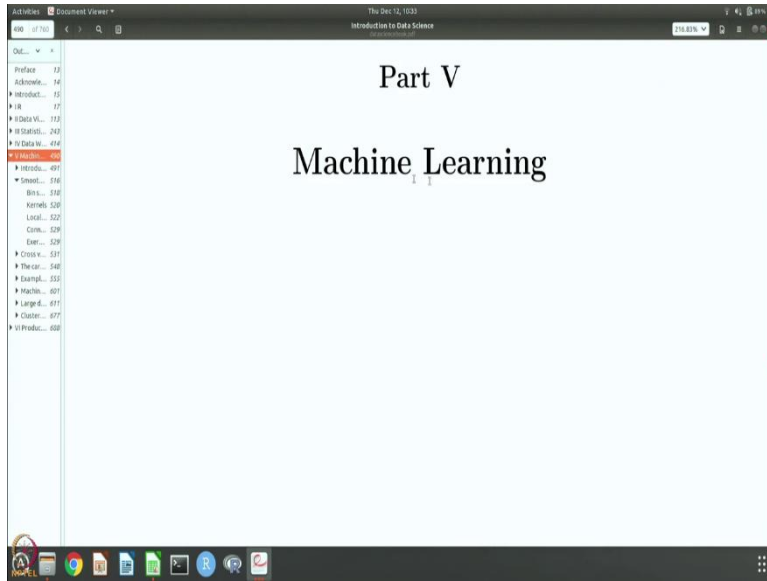
- Raw data: noisy
- Data needs smoothing (remove noise and retain data)
- Negative stress; stress with zero strain
- Data needs clean-up
- How to clean up the data and smooth the data to carry our further analysis?
- Consider the data from stress-strain experiment on Aluminium and Brass
- Task: clean and smooth the data; calculate the modulus and the measures of strength (yield stress, 0.2% proof stress, and ultimate tensile strength)
- Smoothing: leads to machine learning – Irizarry's book on Data Science

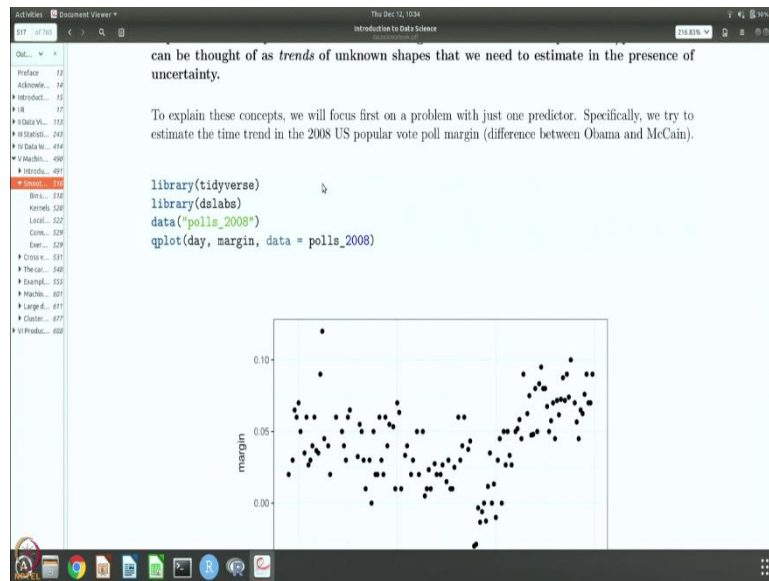
NPTEL Guru and Hina Dealing with Materials Data IITB 4 / 6

So the summary version of what we have seen is that the raw data is noisy so it needs smoothing; that is to remove noise and retain only the data which we are going to do. There are also things like negative stress and stress with zero strain and so on. So this data needs clean-up. How to clean up the data and smooth the data to carry out further analysis is what we are going to do in this case study.

And we are going to use data sets from stress strain experiment on both aluminium and brass and I will do the exercise for aluminium and I will leave the brass data set with you so that you can do same things and see how it works. So our task is to clean and smooth the data, calculate the modulus and a measure of strength, so we want to get and smoothing also leads to machine learning and Irizarry's book on Data Science has this aspect described in detail.

(Refer Slide Time: 5:01)

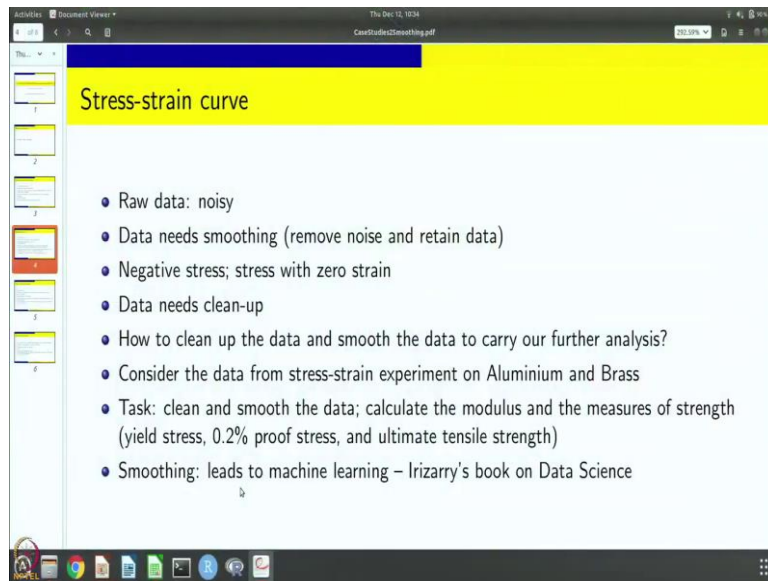




In fact, I strongly recommend that you use this book, like we mentioned long back that it is a book that is feely available and you can see that there is a chapter 29 on Smoothing and it comes in the part on Machine Learning. So, smoothing is technique and it is called curve fitting or low pass filtering. And it is very useful and as you see here so there is a data which has a smooth trend and there is noise and when you add them up instead of looking like this the data looks like this.

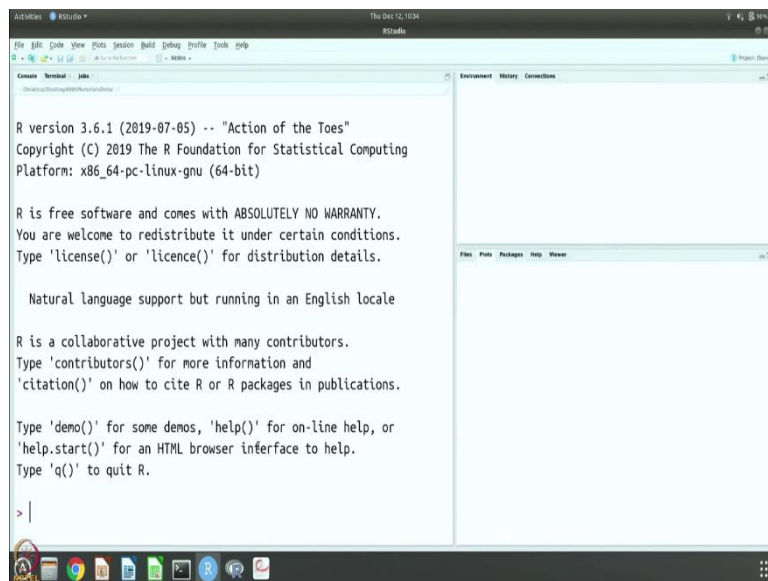
So our aim is to separate out this part and remove it and so get the data back to this kind of curve so that we can do the analysis on that. Okay. So there are several different ways of doing it and we will do one manually ourselves and then we will use some of the commands that is given in this book, but in any case I strongly recommend that you go through Irizarry's book and so it has more information than what we are going to discuss, which might come handy for you when you do smoothing of your own data.

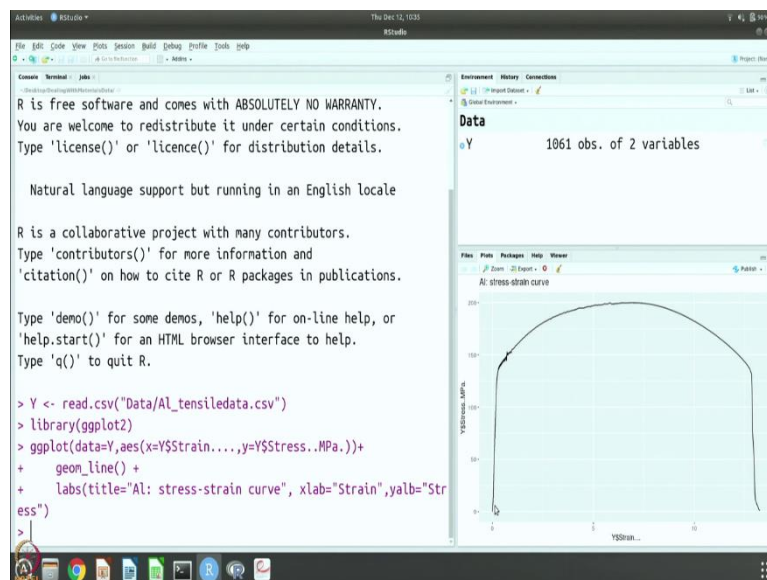
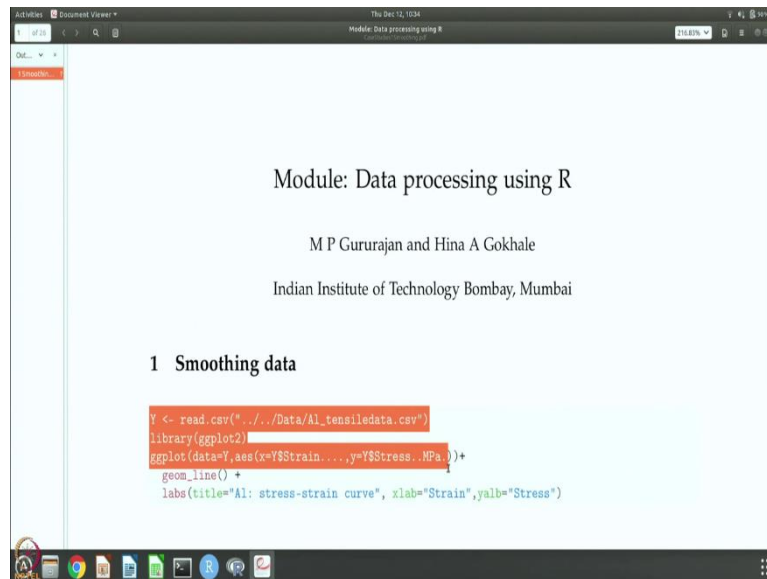
(Refer Slide Time: 6:06)



So let us go back to our data and let us begin with plotting the data.

(Refer Slide Time: 6:13)



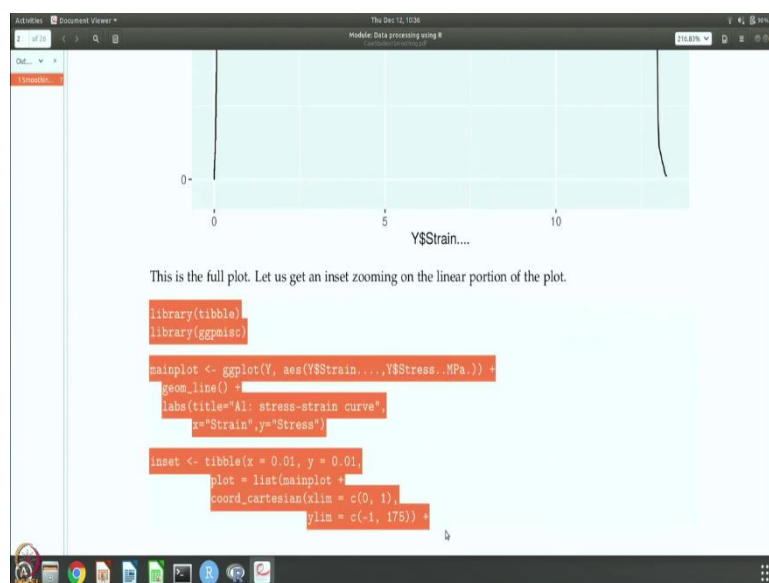
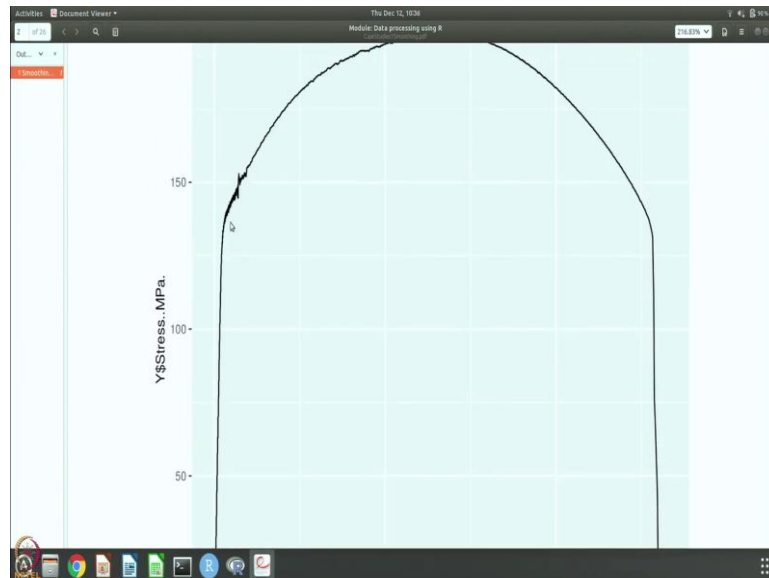


So let us open our and let us do the reading of data and plotting it. Okay, so we have read the data, aluminium tensile data, it is in csv format so we use ggplot and we plot stress versus strain and we collect the points through a line, and we have labelled it as alumina stress strain curve. Now you can see that the data at least here, for example, show some kind of noisy behaviour and there is some noise here also.

And even in this initial part, there is some noise but we are not able to see it clearly because you know when we draw schematic stress strain curves, we draw linear portion and then show the deviation from linearity, but that linear portion is exaggerated to show clearly how it looks, but in most of the materials that portion is very small. Elastic strains are very-very small compared to the total strain so this is a very small portion of the curve.

So we need to basically zoom on this part and show it as an inset to see how this part looks, from where we are going to get the modulus.

(Refer Slide Time: 7:54)



```

inset <- tibble(x = 0.01, y = 0.01)
plot = list(mainplot +
  coord_cartesian(xlim = c(0, 1),
    ylim = c(-1, 175)) +
  )

labs(title=NULL, x = NULL, y = NULL) +
  theme_bw(10)))

mainplot + expand_limits(x = 0, y = 0) +
  geom_plot_npc(data = inset, aes(npcx = x, npcy = y, label=plot))

Al: stress-strain curve

```

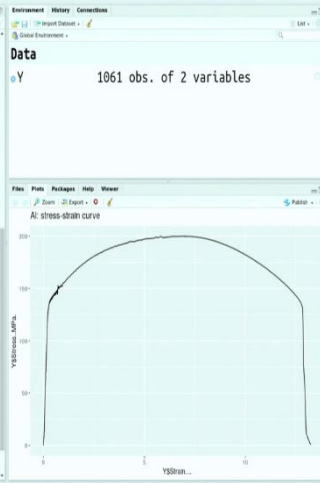
So to do that let's do the next exercise, okay. So here it is seen better so there is noise and here you cannot very clearly see, but there is a little bit of a noise here too. So to do this, we are going to do this exercise.

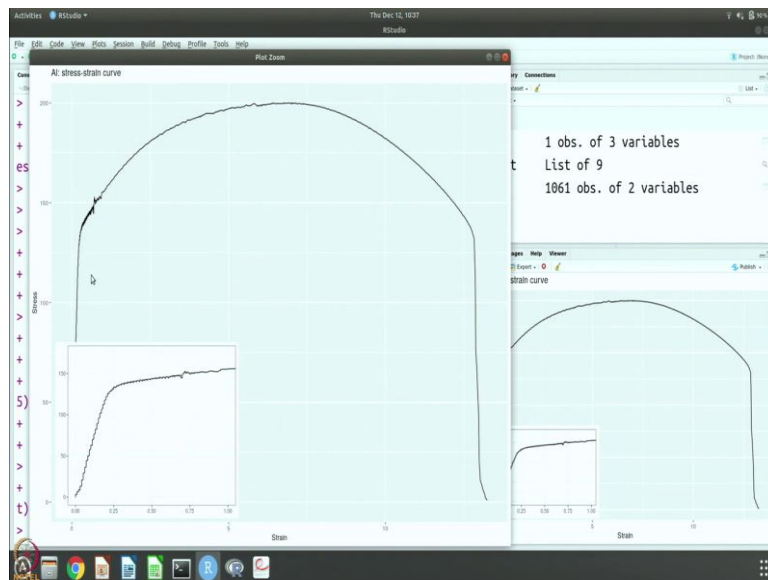
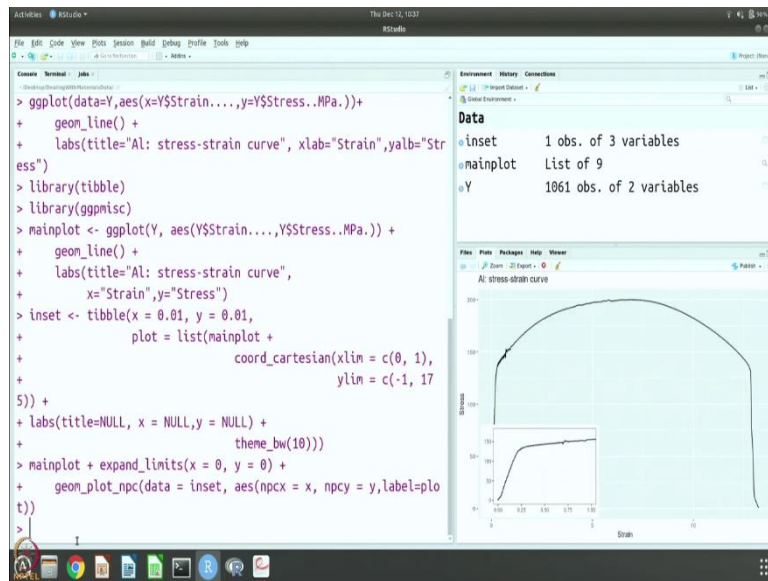
(Refer Slide Time: 8:15)

```

> Y <- read.csv("Data/Al_tensiledata.csv")
> library(ggplot2)
> ggplot(data=Y, aes(x=YSStrain..., y=YSStress..MPa.)) +
  + geom_line() +
  + labs(title="Al: stress-strain curve", xlab="Strain", ylab="Stress")
> library(tibble)
> library(ggpmisc)
mainplot <- ggplot(Y, aes(YSStrain..., YSStress..MPa.)) +
  geom_line() +
  labs(title="Al: stress-strain curve",
    x="Strain", y="Stress")
inset <- tibble(x = 0.01, y = 0.01,
  plot = list(mainplot +
    coord_cartesian(xlim = c(0, 1),
      ylim = c(-1, 175))
  )) +
  labs(title=NULL, x = NULL, y = NULL) +
  theme_bw(10)))
mainplot + expand_limits(x = 0, y = 0) +

```



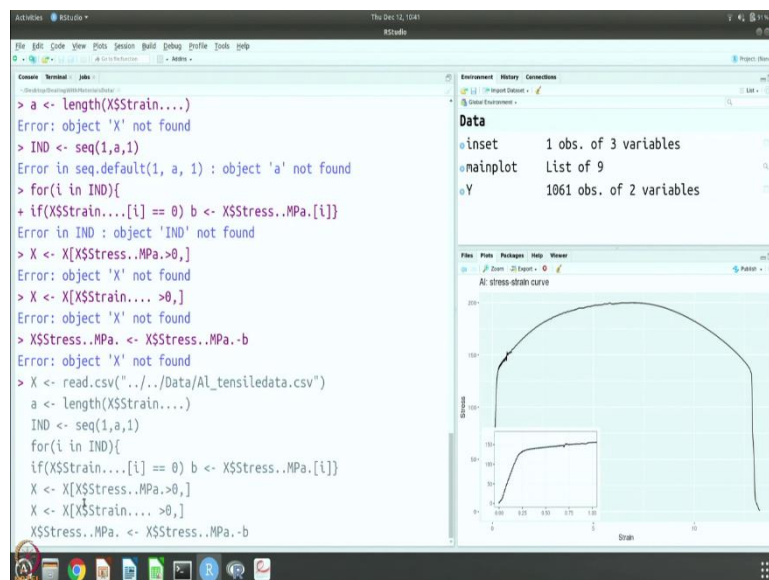
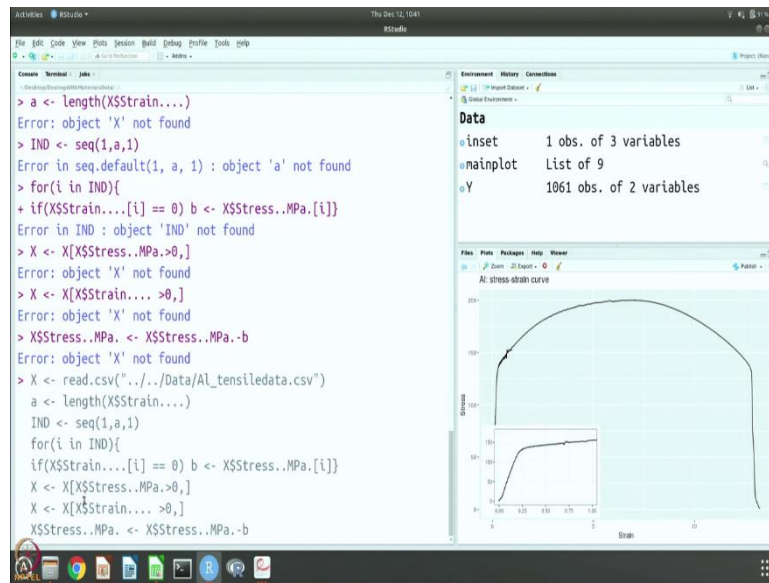


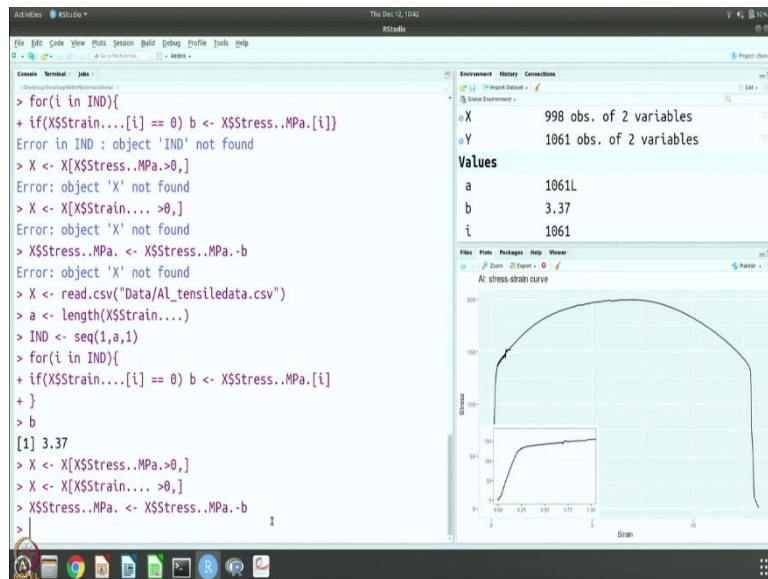
So we are going to use the library table and we are going to use library ggp miscellaneous and so we are going to define the main plot which is the same stress strain curve and we have the data, so we are going to plot the strain stress in the data. Then we are going to prepare inset and the inset gives you the limits for the X and Y axis and from the main plot, it is going to take this portion up the curve and it is going to prepare an inset.

And then we are going to off course plot them together, main plot and with the inset. So that is what this command does. So as you can see, you can see that there is this main curve and we have taken small portion from here and we have expanded and if you zoom in you can see that what looks like a neat straight line here actually also has lots of these wiggles. So what we need is actually a smooth curve and you can see that the initial portion still has some problems because maybe the stress strain measurements are not really perfect here.

So this looks like a straight line, so if you extend it slightly, it should go something like this but it has a different slope and this is a common problem initial when you put the sample in and put the grips on and try to do the experiment initially, maybe there are small adjustments that has to take place before proper loading happens and your measurements of load and displacement are reliable.

(Refer Slide Time: 12:57)





The screenshot shows a LibreOffice Calc spreadsheet with the following data:

Strain	Stress
2.53	0
2.72	0
2.89	0
3.13	0
3.37	0
3.38	0.01
3.37	0.01
3.46	0.01
3.71	0.01
3.9	0.01
4.08	0.01
4.17	0.01
4.11	0.01
4.18	0.01
4.35	0.01
4.54	0.01

So this is the, so we need the clean-up part, we have seen the data itself that it requires cleaning up. We also now see that there is a need for smoothing of the data. So the exercise now is to do both the clean-up and smoothing. So to do that, okay, so let us take this. So we need to read the data and then this gives you the length of the data that is how many data points are there and then we are going to go through each of the data points.

And then what we are going to do? We are going to say that if the strain is zero, we are going to keep track of what is stress. So once you go through all data points, so you will see at the beginning, where it measures strain to be zero, even though the stress is not zero, you will know what is that stress value and because it keeps rewriting to the same value you will know what is the highest stress value for which the strain is marked as zero.

So that value will be stored as B for us. And then we are going to take the data and whatever values which are greater than zero in terms of stress and strain, those are the only things that we are going to consider and we are also going to remove this, so we are going to offset the stress in such a way that it starts at values, when the stress is non-zero the strain will also be non-zero.

So the previous stress value which shows some value, so we are going to subtract it out, so that it starts at 0. So that is what is being done here and let us do this. So you can see that B is 3.37 and from the data also, we see that 3.37 is the value at which it still shows zero. So if you subtract 3.37 from all the stress, you will see that it is 0 and then 0.01, 0.01 and so on and so forth.

So we are going to use this and you can see where the noise comes from, so 3.38 and again it becomes 3.37, so it will give you as 0.01, 0.01 and 0 and 0.01. So, this is the small wiggly thing that you see in the plot. So this part of the data, now if you let us say head x. So, you can see that it gives you only positive and non-zero stress and whenever the stress is non-zero, you also see that the strain is non-zero.

Even though there is a small noise because here it is zero, so it should be zero, but you do see some noise there. So this is the data now we have, using this data now we are going to do the analysis. So the first thing to do is to take out the linear portion of the curve and fit it to a straight line and from the slope of the curve, we can evaluate the modulus. So that is the first exercise we want to do, so let us do that exercise.

(Refer Slide Time: 15:30)

The RStudio console shows the following code and output:

```
Error: object 'X' not found
> X <- read.csv("Data/AL_tensiledata.csv")
> a <- length(X$Strain...)
> IND <- seq(1,a,1)
> for(i in IND){
+ if(X$Strain...[i] == 0) b <- X$Stress..MPa.[i]
+ }
> b
[1] 3.37
> X <- X[X$Stress..MPa.>0,]
> X <- X[X$Strain... >0,]
> X$Stress..MPa. <- X$Stress..MPa.-b
> head(X)
  Stress..MPa. Strain...
64      0.01      0.01
65      0.00      0.00
66      0.09      0.01
67      0.34      0.01
68      0.53      0.01
69      0.71      0.01
```

The Environment pane shows:

- X: 998 obs. of 2 variables
- Y: 1061 obs. of 2 variables
- Values: a = 1061L, b = 3.37, i = 1061

The Plot pane shows a plot titled "A: stress-strain curve" with Stress on the y-axis and Strain on the x-axis. The plot shows a typical stress-strain curve for a material, with an initial linear elastic region, a yield point, a plastic region, and a final fracture point.

```
if(X$Strain...[i] == 0) b <- X$Stress..MPa.[i]
}
b

## [1] 3.37

X <- X[X$Stress..MPa.>0,]
X <- X[X$Strain... >0,]
X$Stress..MPa. <- X$Stress..MPa.-b
a <- length(X$Strain...)
m <- 25
x <- data.frame(matrix(nrow = a-2*m-1, ncol = 2))
colnames(x) <- c("stress", "strain")
INDEX <- seq(m,a-m,1)
j=1
for(I in INDEX){
  p <- I-m+1
  q <- I+m-1
  x[j,1] <- sum(X$Stress..MPa.[p:q])/(2*m-1)
  x[j,2] <- sum(X$Strain...[p:q])/(2*m-1)
  j = j+1
}
ggplot(x,aes(strain, stress))+geom_line()
ggplot(x,aes(strain, stress))+geom_line()+xlim(0,0.2)
```

Warning: Removed 679 rows containing missing values (geom path)

The RStudio console shows the following code and output:

```
64      0.01      0.01
65      0.00      0.01
66      0.09      0.01
67      0.34      0.01
68      0.53      0.01
69      0.71      0.01
> a <- length(X$Strain...)
n <- 25
x <- data.frame(matrix(nrow = a-2*n-1, ncol = 2))
colnames(x) <- c("stress", "strain")
INDEX <- seq(n,a-n,1)
j=1
for(I in INDEX){
  p <- I-n+1
  q <- I+n-1
  x[j,1] <- sum(X$Stress..MPa.[p:q])/(2*n-1)
  x[j,2] <- sum(X$Strain...[p:q])/(2*n-1)
  j = j+1
}
ggplot(x,aes(strain, stress))+geom_line()
ggplot(x,aes(strain, stress))+geom_line()+xlim(0,0.2)
```

The Environment pane shows:

- X: 998 obs. of 2 variables
- Y: 1061 obs. of 2 variables
- Values: a = 1061L, b = 3.37, i = 1061

The Plot pane shows the same stress-strain curve as in the first image.

So how do we do that and here is a code which does the smoothing first. So let us complete the smoothing first. So now because we have edited a data little bit and removed the portions where we had some clean-up and negative stress and things like that, so we have removed them. So we have to get the new size of the data.

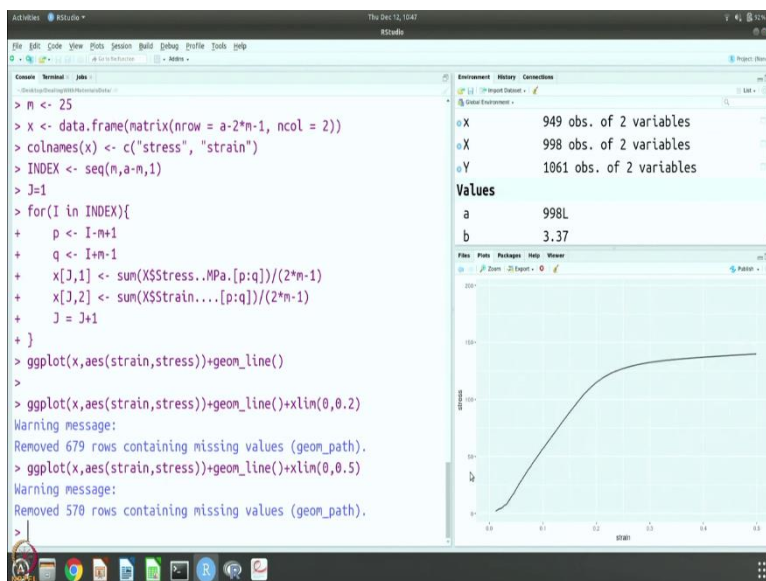
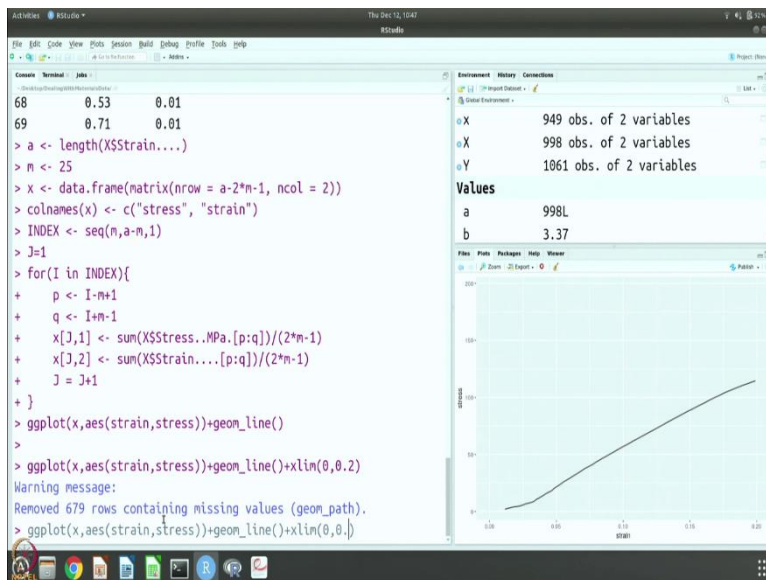
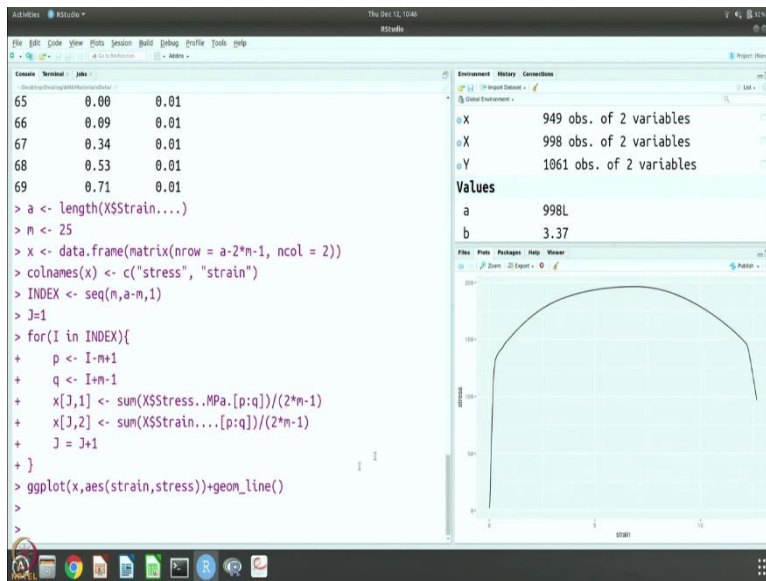
So that is what m is and then we are going to use this m to be 25 that is, this is the size of the box that we are going to use to do the smoothing and this can be different so you can actually play around with it and find out what is the right number which gives you a smoother data and the smoothen data we are going to store in the variable x , so it is a data frame and it has so many data points.

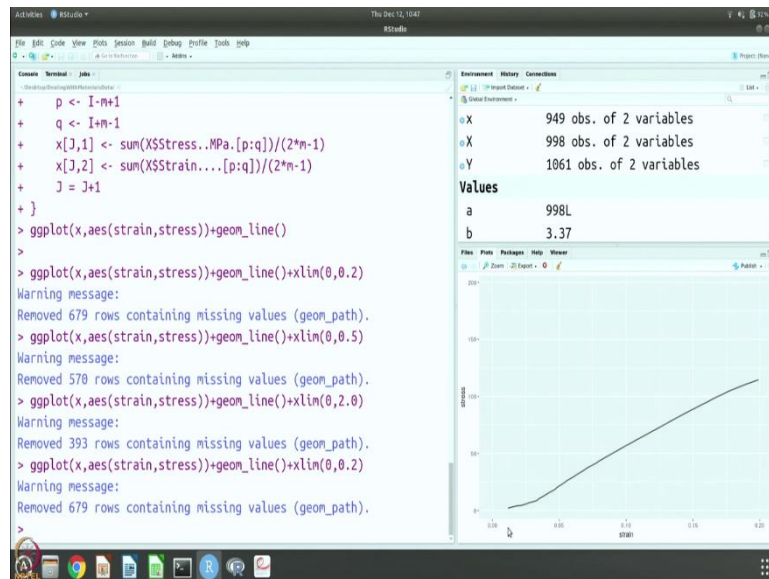
And then it has 2 columns so stress or strain which are smoothened values which are going to be stored here. And index is a sequence, so it starts from 25 and it goes up the length minus 25 and it increases in one. And what we are going to do as you can see here is that we are going to take every data point and we are going to take 25 data points before and 25 data points after.

And we are going to average the data points for stress and strain over these data points that chosen and the averaged out value we are going to store as the stress and strain at that point, right. Suppose if I take i , so index it starts with 25, so 25th point that I will take, then I will take 24 points before that and 24 points after that including the 25th point, so now you have 49 points, so that is where it is $2 \times m - 1$.

So we have averaged this 49 points, stress and divide by that and we are storing it as the stress at that point, which is the 25th data point in the original data set, the cleaned up data set and so we are going to then plot this stress and strain that we get from this smoothen data and then we are also going to see the linear portion of the curve. So let us first plot the, let us first plot the curve, smoothen and plot the curve.

(Refer Slide Time: 18:19)





So you can now clearly see that the data is very nicely smoothened, okay and now you can look at only this portion of the curve which is where I am restricting the x limit to go from 0 to 0.2. So this is 2.5, so we are restricting ourselves to very small, resultant you can see that it is a straight line. So you can also do a little bit, maybe 0.5 or something.

So you can see that this is the, this is why I am plotting up to 0.2 because if you plot up to 0.5, you can see that the curve changes, so you can, for example, let us see the full curve. So this is the curve, right? So up to 2 if you plot, this is the curve. So up to 0.5 if you plot, you see this part and from that curve you know that somewhere around 0.2, 0.25 is where the change is.

And if you plot for the smaller region, you can also see that there is this small portion, so this is the curve and it is extended, it should go like that but the curve goes like this. So there is some small error here and so if you leave this out, the remaining portion is actually the straight line response, which is the linear response. From which one can calculate the modulus.

(Refer Slide Time: 19:49)

```

for(I in INDEX){
  p <- I-m+1
  q <- I+m-1
  x[J,1] <- sum(X$Stress..MPa.[p:q])/(2*m-1)
  x[J,2] <- sum(X$Strain...[p:q])/(2*m-1)
  J = J+1
}
ggplot(x,aes(strain,stress))+geom_line()
ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)

## Warning: Removed 679 rows containing missing values (geom_path).

xs <- x$strain[0:200]
ys <- x$stress[0:200]
plot(xs,ys)
fit <- lm(ys ~ xs)
abline(fit$coefficients,col="red")
plot(fit$residuals)
qqnorm(fit$residuals)
modulus <- fit$coefficients[2]
UIS <- max(x$Stress)
modulus

##          XS
## 658.5707

```

```

+       J = J+1
+     }
+   }
> ggplot(x,aes(strain,stress))+geom_line()
>
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)
Warning message:
Removed 679 rows containing missing values (geom_path).
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.5)
Warning message:
Removed 570 rows containing missing values (geom_path).
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,2,0)
Warning message:
Removed 393 rows containing missing values (geom_path).
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)
Warning message:
Removed 679 rows containing missing values (geom_path).
> xs <- x$strain[0:200]
  ys <- x$stress[0:200]
plot(xs,ys)
fit <- lm(ys ~ xs)
abline(fit$coefficients,tol="red")

```

Environment	History	Connections
X	949 obs. of 2 variables	
X	998 obs. of 2 variables	
Y	1061 obs. of 2 variables	

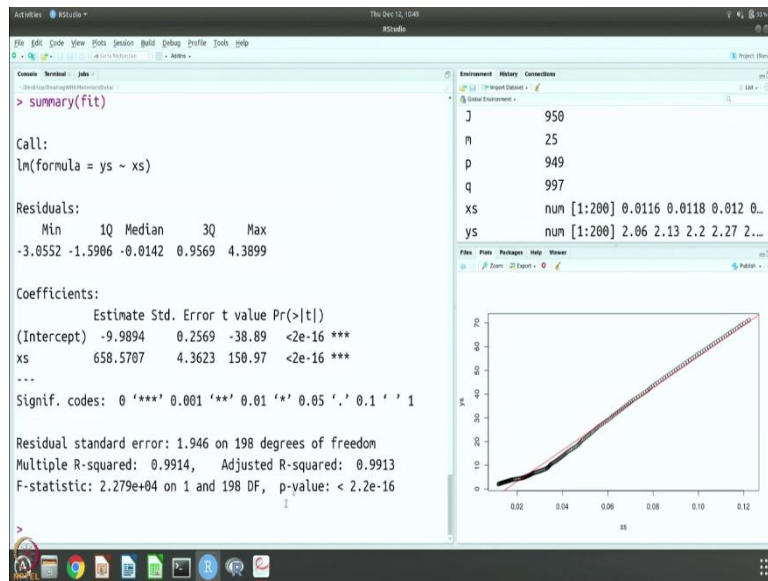
Values	
a	998L
b	3.37

```

+   }
+ }
> ggplot(x,aes(strain,stress))+geom_line()
>
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)
Warning message:
Removed 679 rows containing missing values (geom_path).
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.5)
Warning message:
Removed 570 rows containing missing values (geom_path).
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,2,0)
Warning message:
Removed 393 rows containing missing values (geom_path).
> ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)
Warning message:
Removed 679 rows containing missing values (geom_path).
> xs <- x$strain[0:200]
  ys <- x$stress[0:200]
plot(xs,ys)
fit <- lm(ys ~ xs)
abline(fit$coefficients,col="red")

```

Environment	History	Connections
J	950	
m	25	
p	949	
q	997	
xs	num [1:200] 0.0116 0.0118 0.012 0...	
ys	num [1:200] 2.06 2.13 2.2 2.27 2...	



Stress (MPa)	Strain (%)
-0.15	0
0.01	0
0.03	0
0.27	0
0.62	0
0.85	0
1.29	0
1.54	0
1.6	0
1.33	0
1.21	0
1.17	0
1.38	0
1.66	0
1.91	0

So let us do the modulus calculation. So we are going to first 200 points and take the strain and stress and we are going to plot it and we are also going to fit it for a straight line and we are going to plot the fitted line in red. So you can see that these are data points and these are the, this is the fitted line and obviously there is some small region here which is to be discarded and the rest of the data actually fits very nicely for the straight line.

And so if you look at the fit summary, you find that the slope is 658.57 and if you look at original data, then you realise that the stress was given in MPa, so if you look at value of a 658 MPa, so that is, so it should give you the modulus in GPa. So you can fit the straight line and you can get the modulus value and of course you can also check your fitting by plotting the residuals.

(Refer Slide Time: 22:01)

```
for(l in INDEX){
  p <- l-m+1
  q <- l+m-1
  x[J,1] <- sum(X$Stress..MPa.[p:q])/(2*m-1)
  x[J,2] <- sum(X$Strain...[p:q])/(2*m-1)
  J = J+1
}
ggplot(x,aes(strain,stress))+geom_line()
ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)

## Warning: Removed 679 rows containing missing values (geom_path).

xs <- x$strain[0:200]
ys <- x$stress[0:200]
plot(xs,ys)
fit <- lm(ys ~ xs)
abline(fit$coefficients,col="red")
plot(fit$residuals)
qqnorm(fit$residuals)
modulus <- fit$coefficients[2]
UTS <- max(x$stress)
modulus

##      xs
## 658.5707
```

The RStudio interface shows the following output in the console:

```
Call:
lm(formula = ys ~ xs)

Residuals:
    Min       1Q   Median       3Q      Max
-3.0552 -1.5906 -0.0142  0.9569  4.3899

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.9894     0.2569  -38.89  <2e-16 ***
xs           658.5707     4.3623  150.97  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.946 on 198 degrees of freedom
Multiple R-squared:  0.9914, Adjusted R-squared:  0.9913
F-statistic: 2.279e+04 on 1 and 198 DF, p-value: < 2.2e-16

> plot(fit$residuals)
> qqnorm(fit$residuals)
```

The Environment window shows the following variables:

Variable	Value
J	950
n	25
p	949
q	997
xs	num [1:200] 0.0116 0.0118 0.012 0...
ys	num [1:200] 2.06 2.13 2.2 2.27 2...

The Normal Q-Q Plot shows Sample Quantiles on the y-axis (ranging from -3 to 3) and Theoretical Quantiles on the x-axis (ranging from -3 to 3). The data points deviate significantly from the diagonal line, indicating non-random residuals.

So you can plot the residuals and see and of course the residuals are not looking like randomly distributed, so there seems to be some methodical errors and you can also do the Q-Q norm to check if the error is random. It does not seem to be, it is really not a straight line, so there seems to be some deviation from linearity, in any case. So we have the data and we can also calculate the other quantities.

(Refer Slide Time: 22:45)


```

for(I in INDEX){
  p <- I-m+1
  q <- I+m-1
  x[J,1] <- sum(X$Stress..MPa.[p:q])/(2*m-1)
  x[J,2] <- sum(X$Strain...[p:q])/(2*m-1)
  J = J+1
}
ggplot(x,aes(strain,stress))+geom_line()
ggplot(x,aes(strain,stress))+geom_line()+xlim(0,0.2)

## Warning: Removed 679 rows containing missing values (geom_path).

xs <- x$strain[0:200]
ys <- x$stress[0:200]
plot(xs,ys)
fit <- lm(ys ~ xs)
abline(fit$coefficients,col="red")
plot(fit$residuals)
qnorm(fit$residuals)
modulus<- fit$coefficients[2]
UTS <- max(x$stress)
modulus

##          xs
## 658.5707

```

Activities RStudio

File Edit Code view Plots Session Build Debug Profile Tools Help

Console Terminal Jobs

```

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.9894    0.2569  -38.89 <2e-16 ***
xs           658.5707    4.3623  150.97 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.946 on 198 degrees of freedom
Multiple R-squared:  0.9914,    Adjusted R-squared:  0.9913
F-statistic: 2.279e+04 on 1 and 198 DF, p-value: < 2.2e-16

> plot(fit$residuals)
> qqnorm(fit$residuals)
> modulus<- fit$coefficients[2]
> UTS <- max(x$stress)
> modulus
      xs
658.5707
> UTS
[1] 196.2994

```

Environment History Connections

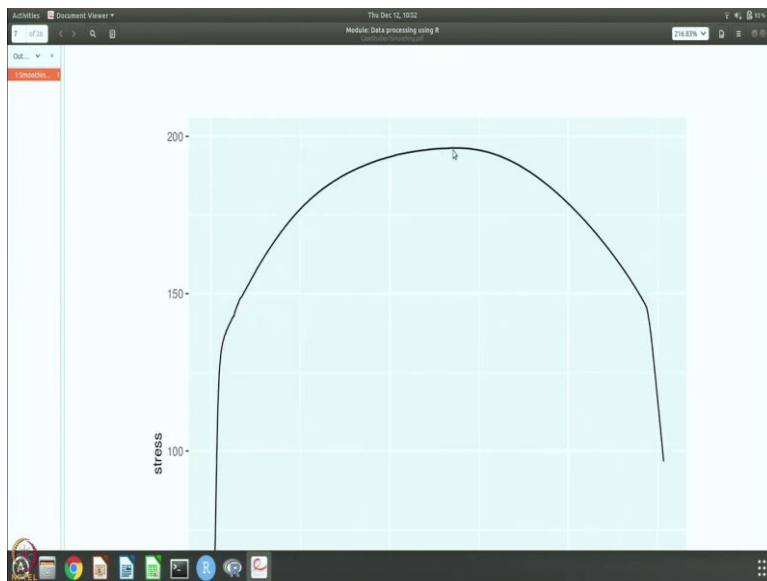
```

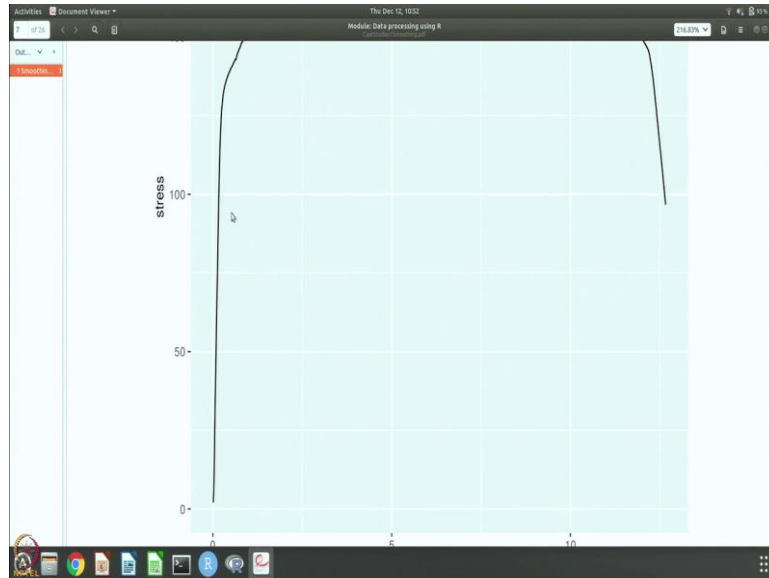
modulus Named num 659
p        949
q        997
UTS     196.299387755102
xs      num [1:200] 0.0116 0.0118 0.012 0...
ys      num [1:200] 2.06 2.13 2.2 2.27 2...

```

Files Plots Packages Help Viewer

Normal Q-Q Plot





For example, so the modulus is nothing but the slope that we have calculated and UTS value is basically the maximum in the stress, so you can calculate the modulus 658, so that is the 65.8 GPa and UTS is 196 MPa. So you can see that we have got values and if you look at the stress strain curve that we plotted sometime back, you will see that, so it is about, UTS is about 200 and we can go and look at the plots and you can see that the UTS is about 200. And the slope of this initial portion of the curve happens to be 658, so that is 65.8 GPa.

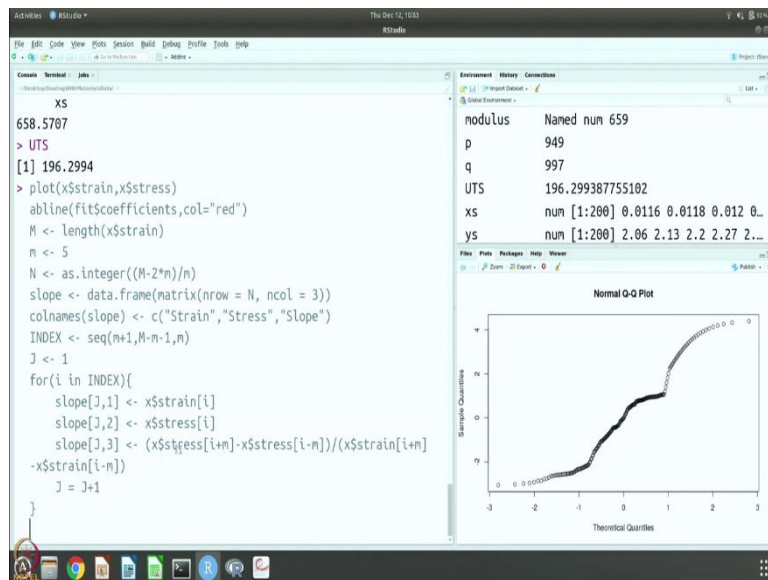
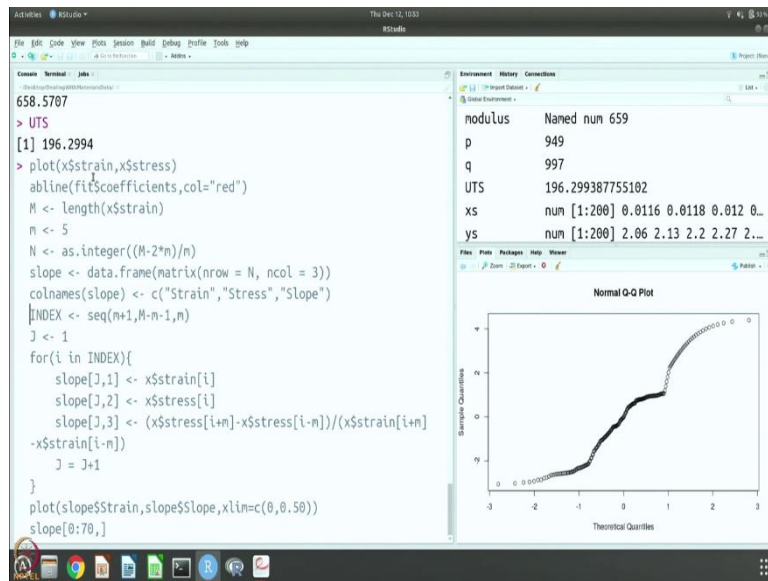
(Refer Slide Time: 24:24)

```

INDEX <- seq(m+1,M-m-1,m)
j <- 1
for(i in INDEX){
  slope[j,1] <- x$strain[i]
  slope[j,2] <- x$stress[i]
  slope[j,3] <- (x$stress[i+m]-x$stress[i-m])/(x$strain[i+m]-x$strain[i-m])
  j = j+1
}
plot(slope$strain,slope$Slope,xlim=c(0,0.50))
slope[0:70,]

##      Strain   Stress   Slope
## 1  0.01265306  2.407959 327.80000
## 2  0.01367347  2.730408 305.60000
## 3  0.01469388  3.031633 290.50000
## 4  0.01571429  3.323265 287.50000
## 5  0.01673469  3.618367 295.50000

```

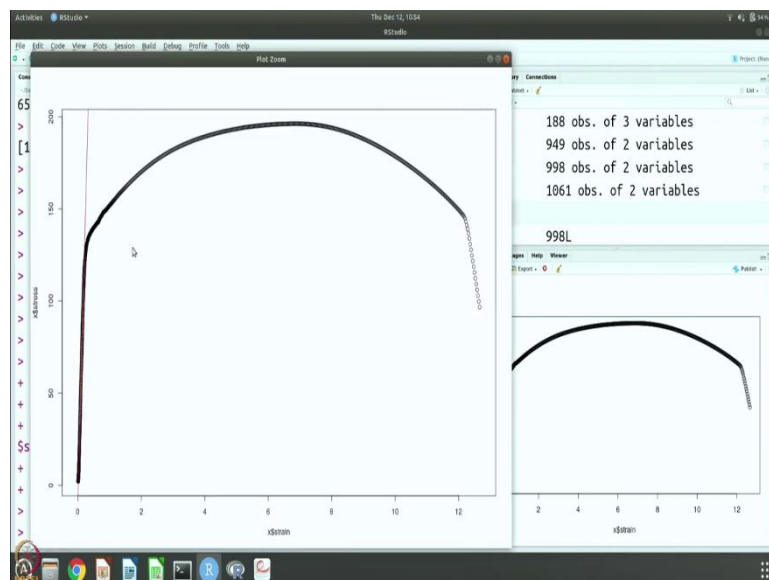
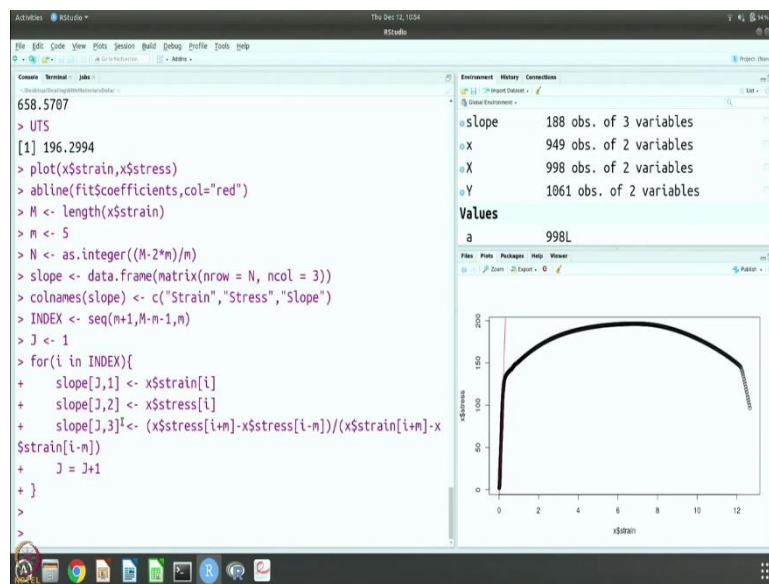



So you can carry out further analysis. What is the other analysis? So let us do one more thing. So what we are trying to do now is we are going to plot the stress versus strain and we are going to draw a line which with the coefficient, so which will be in red. So you will see the stress strain curve with a line drawn, right? So let us do that, let us remove this plot so that we will not be distracted by it.

And then what we are going to do? We are going to calculate the x, what is the length of it and then we are going to take the data points in 5 and we are going to calculate the slope and store it in the variable called slope. So we are going to take the full data and we are going to slide a box of a length 5 and a bin of length 5 and using this, we are going to calculate the slope and that is what this portion of the curve does.

And so it is going to store in data frame called slope, the strain stress and the slope at the given strain value and the given stress value. So that is what this quantity is calculating. And so this is calculating using just a simple difference, so you take the i th point, plus m and minus m you take the difference and divide by the strain. So it is dy by dx and that I am calling as slope.

(Refer Slide Time: 26:14)



So let us do this exercise. Okay, so you can see that this is the stress strain data which is smoothed and this is the line that we have fit, so one measure of the strength you can already see, the deviation from ill strength happens somewhere around 130 that you can see very clearly and so you can say that ill strength is 130.

We have already calculated UTs and that is somewhere around 196 or something. And from this plot itself you can make out that this is where the slope change happens, but you can also do this using our other calculations that we have done in terms of slope and let us do that to know what happens.

(Refer Slide Time: 27:03)

```
INDEX <- seq(m+1,M-m-1,m)
J <- 1
for(i in INDEX){
  slope[J,1] <- x$strain[i]
  slope[J,2] <- x$stress[i]
  slope[J,3] <- (x$stress[i+m]-x$stress[i-m])/(x$strain[i+m]-x$strain[i-m])
  J = J+1
}
plot(slope$strain,slope$Slope,xlim=c(0,0.50))
slope[0:70,]
```

##	Strain	Stress	Slope
## 1	0.01265306	2.407959	327.80000
## 2	0.01367347	2.730408	305.60000
## 3	0.01469388	3.031633	290.50000
## 4	0.01571429	3.323265	287.50000
## 5	0.01673469	3.618367	295.50000

Console Terminal Jobs

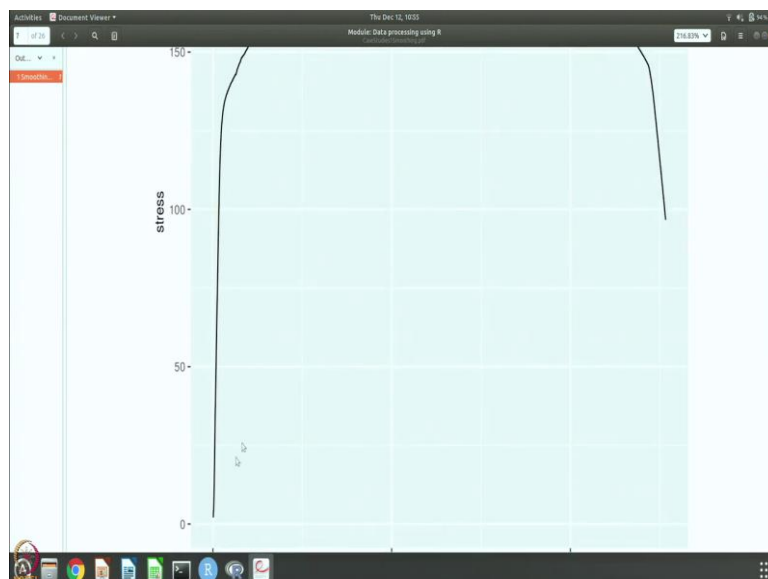
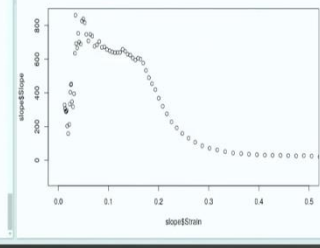
```
51 0.18081633 106.481020 490.18644
52 0.18693878 109.416735 453.85714
53 0.19367347 112.316327 419.16667
54 0.20040816 115.062653 368.13889
55 0.20836735 117.725714 320.11392
56 0.21653061 120.223673 275.29070
57 0.22591837 122.557347 227.15464
58 0.23632653 124.720408 192.05660
59 0.24755102 126.712041 159.11207
60 0.26000000 128.487143 131.03200
61 0.27306122 130.054694 106.72932
62 0.28714286 131.384802 85.78723
63 0.30183673 132.523265 71.36735
64 0.31714286 133.525102 60.89474
65 0.33285714 134.412245 50.69677
66 0.34877551 135.128776 43.00637
67 0.36489796 135.790204 39.26452
68 0.38040816 136.370816 35.55195
69 0.39632653 136.907551 32.02581
70 0.41204082 137.409184 30.66813
```

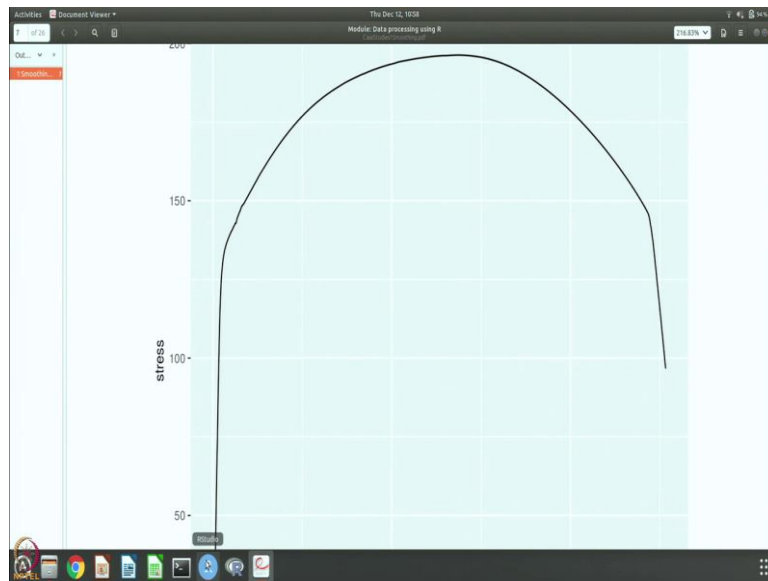
Environment History Connections

- slope 188 obs. of 3 variables
- X 949 obs. of 2 variables
- X 998 obs. of 2 variables
- Y 1061 obs. of 2 variables

Values

a 998L





So let us plot the slope and also list out the first 70 values to know how it looks. So you can see that initially there is some error and there is a constant value and then there is a slope change and then it becomes another constant value here. So from the stress strain curve, it is clear what these values correspond to because if you look at it, so there is some initial problem and then there is a constant value more or less and then there is a change.

And once the change takes place and this portion again you can consider as sort of linear and so it will show you, this is a much deeper curve, it will show you much smaller but constant slope and that is what is being shown, so there is initial some transients and then there is some constant value and then there is a changeover and then there is becoming. So somewhere around 0.35 or 0.4 is where this other slope is coming in.

So this is the transition region and we know that the linear limit is somewhere till 0.12 or something and then there is a changeover and here you can see about 0.4, you get the changed slope, so it becomes plastic. Now here, in this data now you can see the slope is at different strains, it is plotted and you can see that, so there is initial trans entry 327, 290, 203, 158, et cetera.

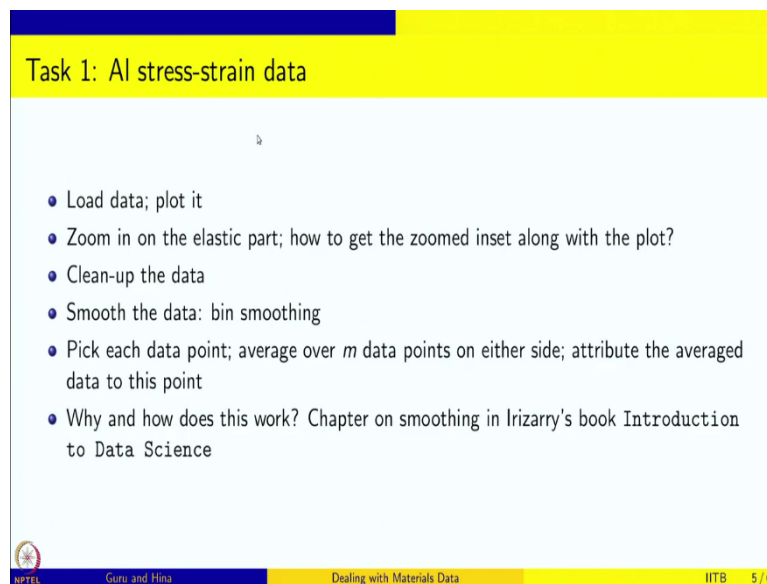
And then it reaches sort of constant value here. We know that it should hover around 680 and that is what it is doing. And after that of course there is a change that happens and somewhere around 0.3, right, is where this becomes sort of straight line 0.3 to 0.4, we can see 0.3 to 0.4, the value changes and about 0.34, 0.36, 0.38, 0.39, 0.4, so it becomes sort of constant value here.

And you can see that at that point, where the slope change has happened, the value of stress is 132, which is what we also saw from the plot that the value, the yield stress is this. Of course, there is one more way to calculate the yield stress which is to look at the take line which is parallel to this line but starts from 0.2 percent and then wherever it intersects with this curve is also the 0.2 percent proved stress.

But we are going to leave that as an exercise for you to do. So this script now shows how to take the data, how to clean up the data, how to smoothen the data and once you have smoothen the data, you can do fitting and which is also something that we have learned. And from the fitting parameters you can evaluate the quantities.

And you also know the error now in the fitting parameters using which you will also be able to tell to what extent is your parameter that you are estimating, namely in this case the modulus, what is a error in the modulus estimation, that also you will get from the fitting exercise. And then you can do other things like measuring the UTS and measuring the deviation from linearity which indicates the onset of plasticity and so one and so forth.

(Refer Slide Time: 31:03)



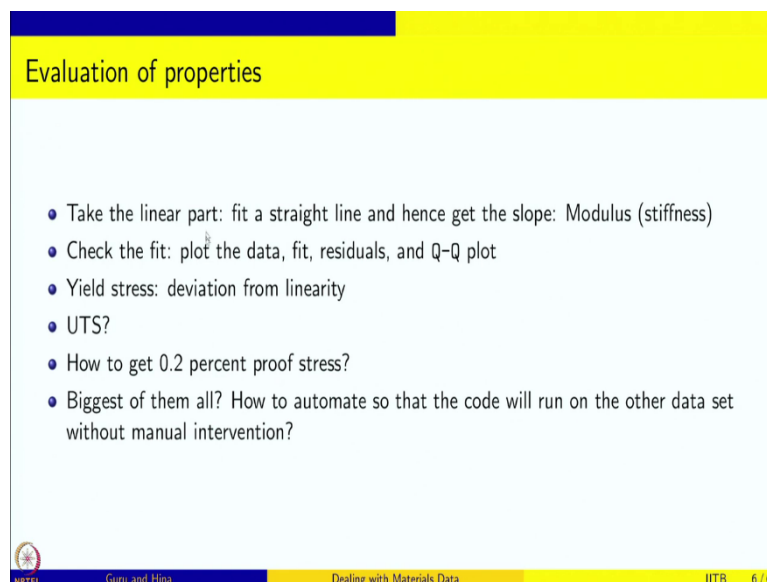
Task 1: AI stress-strain data

- Load data; plot it
- Zoom in on the elastic part; how to get the zoomed inset along with the plot?
- Clean-up the data
- Smooth the data: bin smoothing
- Pick each data point; average over m data points on either side; attribute the averaged data to this point
- Why and how does this work? Chapter on smoothing in Irizarry's book Introduction to Data Science

NPTEL Guru and Hina Dealing with Materials Data IITB 5 / 6

So we will also leave the brass tensile data, the biggest challenge that one faces is that, let us take a look at this. So you can load it, plot it, you can zoom in on the elastic part and you can clean up, smooth, and you can pick each data point and average and that is how smoothing is done.

(Refer Slide Time: 31:16)



Evaluation of properties

- Take the linear part: fit a straight line and hence get the slope: Modulus (stiffness)
- Check the fit: plot the data, fit, residuals, and Q-Q plot
- Yield stress: deviation from linearity
- UTS?
- How to get 0.2 percent proof stress?
- Biggest of them all? How to automate so that the code will run on the other data set without manual intervention?

NPTEL Guru and Hina Dealing with Materials Data IITB 6 / 6

And you can take the linear part to fit a straight line and get the slope so that is a modulus, you can check the fit by plotting the data and fit and residuals and Q-Q etc. You can find the yield stress, you can get UTS. But how do you automate it so that you know if I give some other different code, data, the same code will work and give me the values. Now that is challenging

because we have used some values like for averaging views, the bin size of 25 to smooth and for getting the slope who used a bin size of 5.

Now, how do I do all these things in an automated fashion so that the value comes to me directly? That is a harder problem, but one that I will leave you to play with and we will put both the data sets, so that you can reproduce the results that I have shown as well as do it on brass for your search. Thank you.