Dealing with Material Data: Collection, Analysis and Interpretation

Professor M P Gururajan

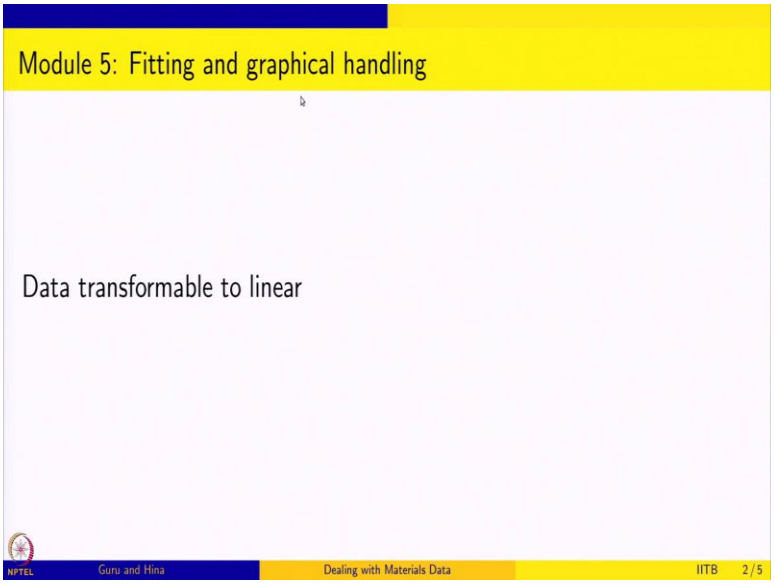
Department of Metallurgical Engineering and Materials Science

Indian Institute of Technology, Bombay

Lecture 76

Data Transformable to Linear

(Refer Slide Time: 00:26)



Module 5: Fitting and graphical handling

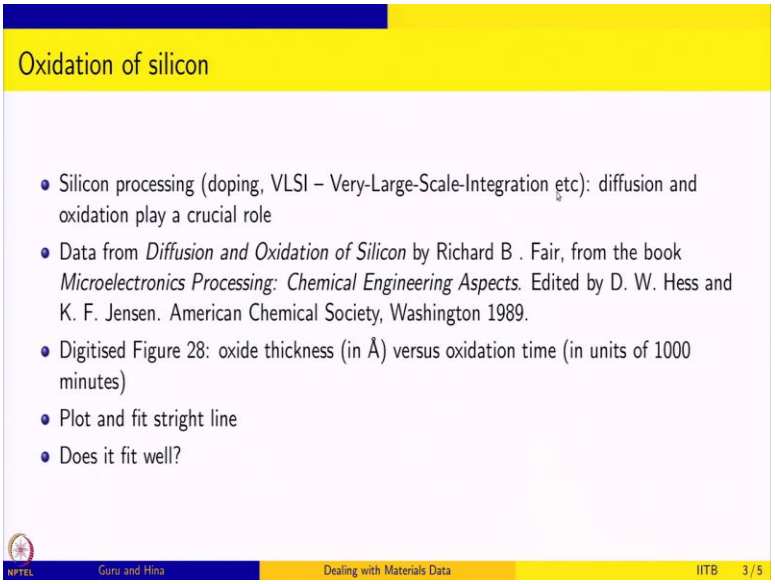
Data transformable to linear

Welcome to the course on Dealing with Materials Data. We are looking at the Collection Analysis and Interpretation of Data from Material Science and Engineering. We are in the fifth module, this is the module on fitting and graphical handling of data. We have already looked at data which is linear and how to fit a linear curve. We also looked at a case where the curve was not linear, but it had a quadratic relationship and we learned how to fit that.

And because fitting is very important and many students who might not be working with statistical methods would still end up doing fitting, because every time you do an experiment and you want to have a relationship between your independent variables and the measurements you are making you do fitting and most of the times the fitting is also called statistical model, so sometimes they are called models. So, you have a set of input data and you model what the output is going to be using which for example if you are given a new set of input data you can predict what the outcome is going to be.

So, it is very important exercise. So, we are going to do more number of practise problems with fitting. And so you will find that this is very useful, even if you do not use the other statistical tools, regression and fitting is something that you would always end up using. So, we are going to consider two cases in this session, in which we are going to consider data, which is not linear, but it is transformable to linear form. So, how to do the fitting? So, you just have to transform it to linear form and then we know how to deal with linear fitting, so that is what we are going to do.

(Refer Slide Time: 02:08)



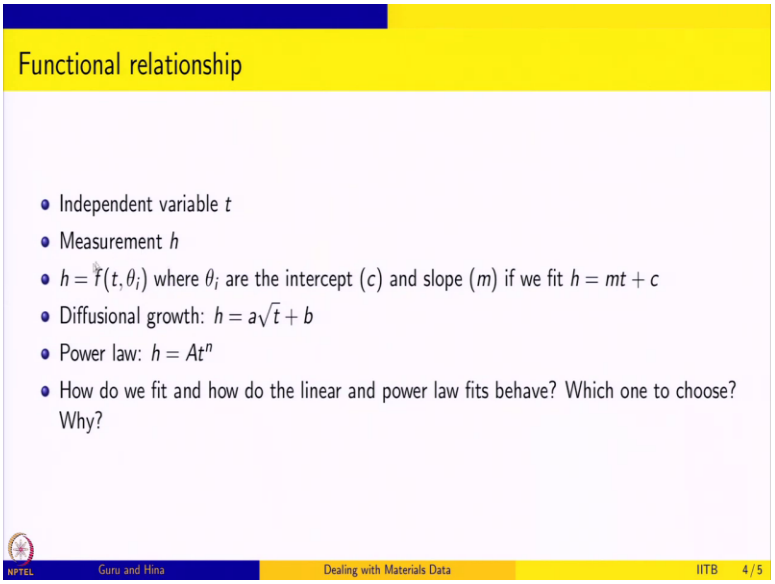
Oxidation of silicon

- Silicon processing (doping, VLSI – Very-Large-Scale-Integration etc): diffusion and oxidation play a crucial role
- Data from *Diffusion and Oxidation of Silicon* by Richard B . Fair, from the book *Microelectronics Processing: Chemical Engineering Aspects*. Edited by D. W. Hess and K. F. Jensen. American Chemical Society, Washington 1989.
- Digitised Figure 28: oxide thickness (in Å) versus oxidation time (in units of 1000 minutes)
- Plot and fit stright line
- Does it fit well?

Specifically, we are going to look at oxidation of silicon. In silicon processing, it is doping or very large scale integration et cetera, the processing involves a diffusion and oxidation, so it is very important to know how diffusion takes place or how oxidation takes place. And so there are lots of processing studies that you would find where people have looked at diffusion of impurities in silicon or oxygen diffusion in silicon and how it affects properties and so on and so forth.

So, this is a data taken from diffusion and oxidation of silicon by Richard B. Fair, this is from the book Micro Electronics Processing: Chemical Engineering Aspects edited by Hess and Jensen, published by American Chemical Society. Like we discussed earlier, the data is not available in tabulated form, so it is in the form of a curve, so I digitized figure 28 of this paper and have made a csv file that consists of oxide thickness in angstroms versus oxidation time and the time is given in units of 1000 minutes. So, we are going to take the data and we are going to plot and we are going to fit a straight line, we will see whether it fits well.

(Refer Slide Time: 03:24)



Functional relationship

- Independent variable t
- Measurement h
- $h = f(t, \theta_i)$ where θ_i are the intercept (c) and slope (m) if we fit $h = mt + c$
- Diffusional growth: $h = a\sqrt{t} + b$
- Power law: $h = At^n$
- How do we fit and how do the linear and power law fits behave? Which one to choose? Why?

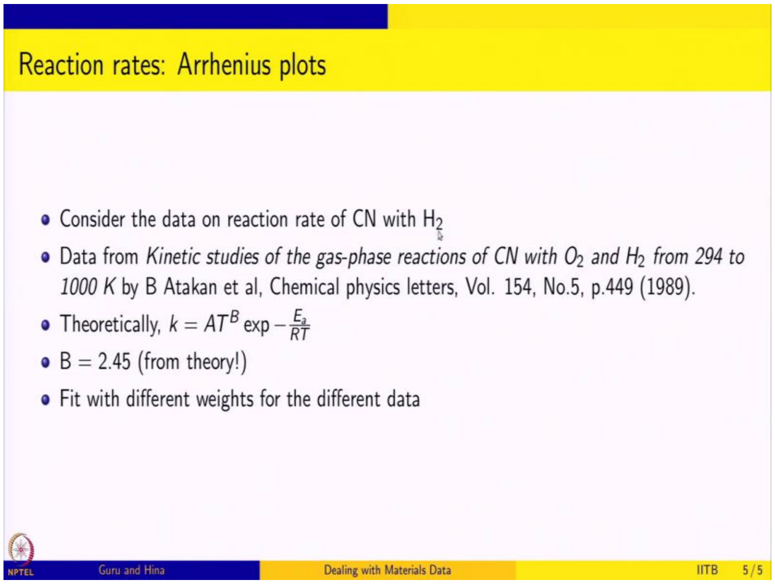
And the independent variable is of course the time, so we are going to vary that and we are going to measure the oxide thickness. So, you can say that the thickness depends on time and what is the functional form and that depends on the parameters that we have to find out, in this case it is the intercept and slope. So, if you say that the thickness is related to

$$h = a\sqrt{t} + b$$

$$h = At^n$$

So, we are going to explore these different fittings and we are going to decide which one to choose, I mean do we choose this or do we choose that and why and so those are the questions that one is interested in. So, this is the first exercise we will do oxidation of silicon.

(Refer Slide Time: 04:35)



Reaction rates: Arrhenius plots

- Consider the data on reaction rate of CN with H_2
- Data from *Kinetic studies of the gas-phase reactions of CN with O_2 and H_2 from 294 to 1000 K* by B Atakan et al, Chemical physics letters, Vol. 154, No.5, p.449 (1989).
- Theoretically, $k = AT^B \exp - \frac{E_a}{RT}$
- B = 2.45 (from theory!)
- Fit with different weights for the different data

The next exercise is on the reaction rate of CN with H2, this is a gas phase reaction and so it is from the paper kinetic studies of gas phase reactions of CN with O2 and H2 from 294 to 1000 Kelvin by Atakan et al and this is from chemical physics letters. The data is given in tabular form in this case and generally these reactions are supposed to be Arrhenius type relations.

$$k = AT^B \exp(-\frac{E}{RT})$$

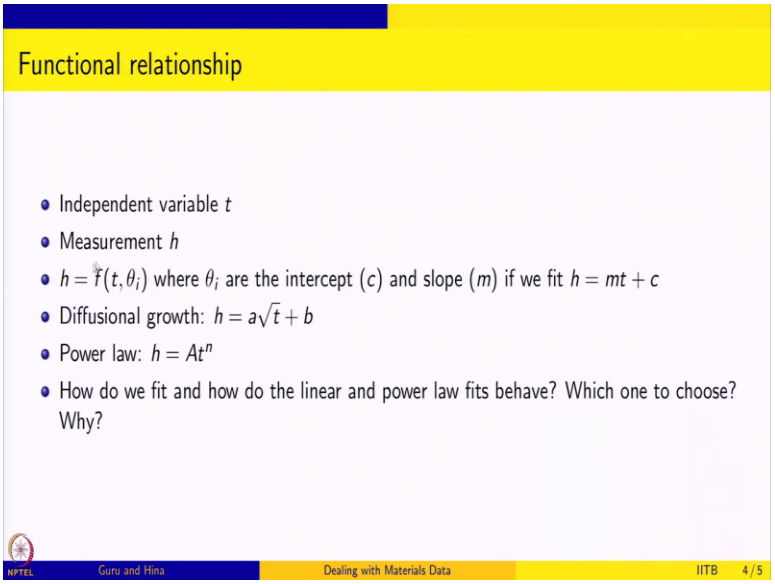
And it is also known from theory that this B value is 2.45, so one thing that you can do is you can take the data you can just directly fit it and find out what is the B value that you get and if you find that the B value is not 2.45, you can try to fit it by taking T to the power 2.45 and find out A and E values.

And in this case, I think they have also done fitting by giving different weights, statistical weights to different data points, I will show you why and there is also an interesting problem, because you will see that different types of fitting gives you different numbers and the paper does not give too many details about how the fitting is done.

And it just says okay we fit and then they give you there are also error that is mentioned in different parts, so it is slightly involved, so we are going to do only partly and I recommend that you go through this paper explore for yourself and try to understand everything that is happening and so that kind of exercise will help you do more of this kind of analysis for yourself.

So, this is again, I mean reaction rates are typically Arrhenius, so this is an important part, because this is in this form, you can see that by taking logarithm we can turn it into a linear form. In this case, it is not quite linear because it is T to the power something and T to the power minus 1 when you take logarithm, so you will have log T and 1 by T, so that is going to be the form.

(Refer Slide Time: 07:08)



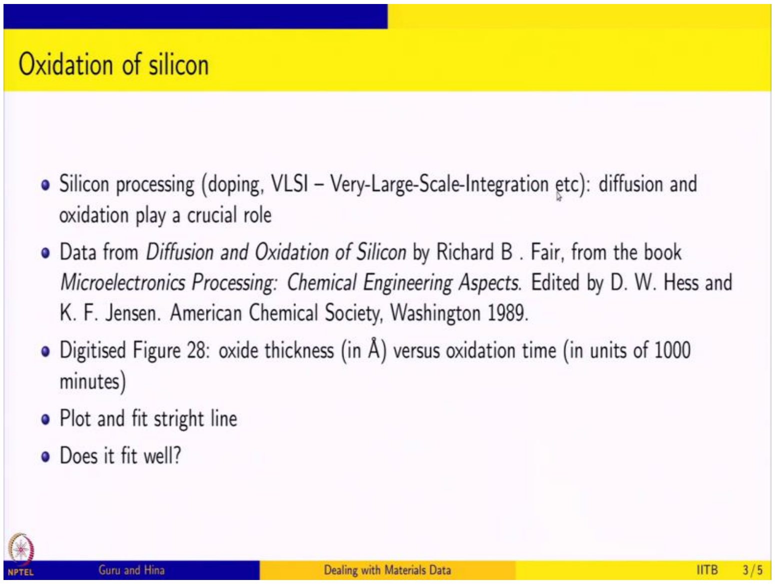
Functional relationship

- Independent variable t
- Measurement h
- $h = f(t, \theta_i)$ where θ_i are the intercept (c) and slope (m) if we fit $h = mt + c$
- Diffusional growth: $h = a\sqrt{t} + b$
- Power law: $h = At^n$
- How do we fit and how do the linear and power law fits behave? Which one to choose? Why?

And in the other case again, it is turned into linear if you just take logarithm, but if it is this it is linear already and here again, you can make it linear if you consider root t instead of t and try to do the calculations. So, these are some of the forms like power law forms can be turned into linear things that come into exponential can be turned into linear by taking logarithms.

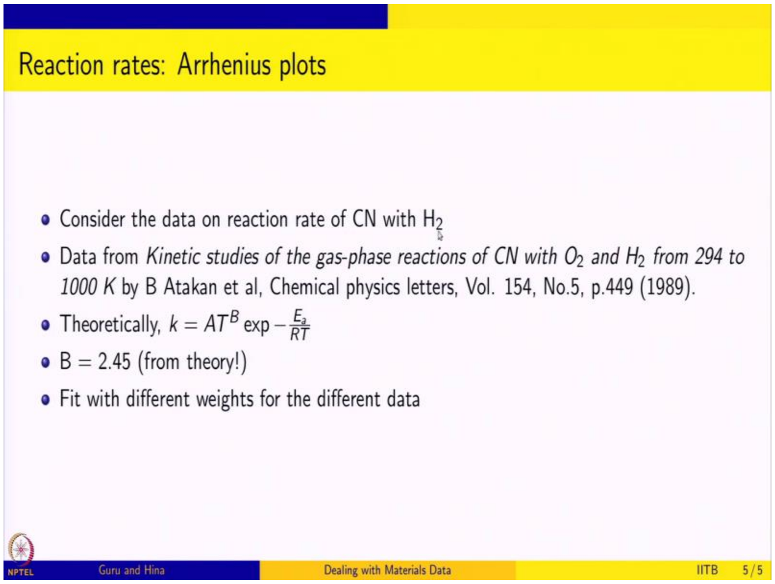
And in other cases by just replacing the variables by a transformed variable again you can make it into a linear form. So, if you can turn it into linear form then how do you deal with it that is already known to us, so we can just do a linear model fitting to the data, so that is what we are going to do with these two data sets.

(Refer Slide Time: 07:56)



Oxidation of silicon

- Silicon processing (doping, VLSI – Very-Large-Scale-Integration etc): diffusion and oxidation play a crucial role
- Data from *Diffusion and Oxidation of Silicon* by Richard B . Fair, from the book *Microelectronics Processing: Chemical Engineering Aspects*. Edited by D. W. Hess and K. F. Jensen. American Chemical Society, Washington 1989.
- Digitised Figure 28: oxide thickness (in Å) versus oxidation time (in units of 1000 minutes)
- Plot and fit stright line
- Does it fit well?

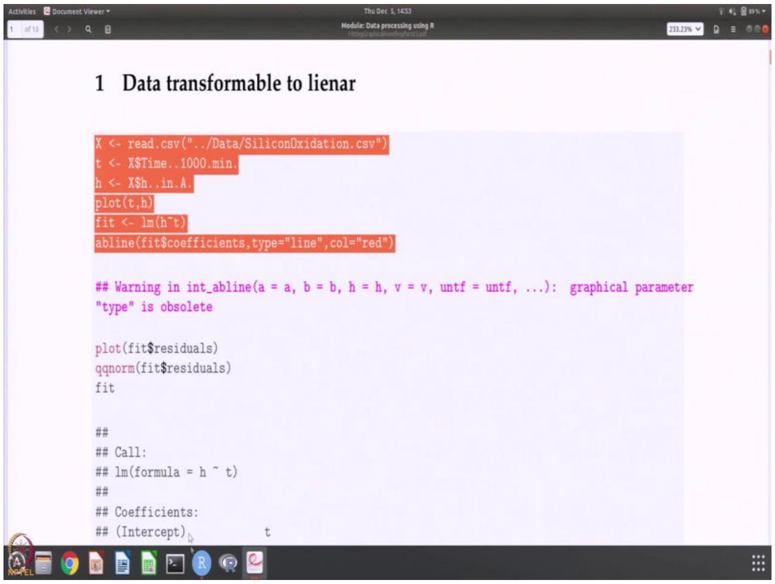


Reaction rates: Arrhenius plots

- Consider the data on reaction rate of CN with H_2
- Data from *Kinetic studies of the gas-phase reactions of CN with O_2 and H_2 from 294 to 1000 K* by B Atakan et al, Chemical physics letters, Vol. 154, No.5, p.449 (1989).
- Theoretically, $k = AT^B \exp -\frac{E_a}{RT}$
- B = 2.45 (from theory!)
- Fit with different weights for the different data

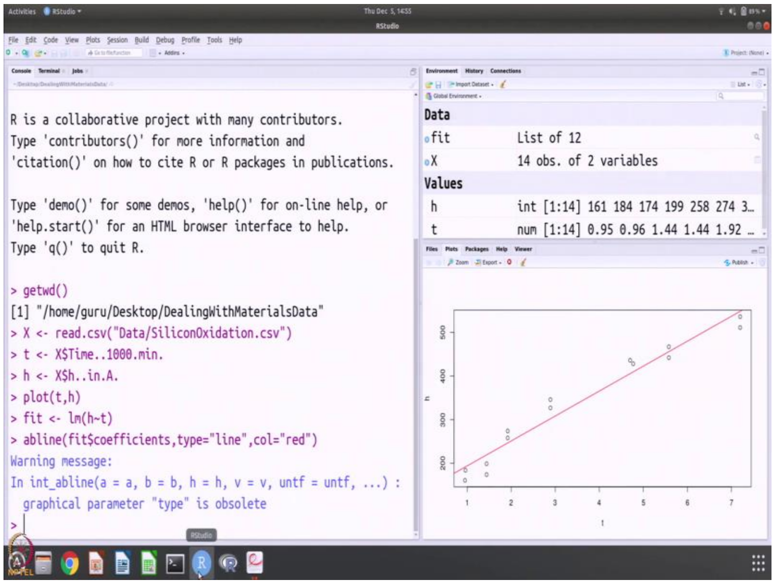
One is for the oxidation data on silicon, one is the oxidation data on silicon. The second is the gas phase reaction of cyanide with oxygen and hydrogen. We are going to specifically look at cyanide hydrogen data and we are going to understand how the reaction rate the k is related to the temperature, so that is what we are going to fit and see.

(Refer Slide Time: 08:31)



1 Data transformable to lienar

```
X <- read.csv("../Data/SiliconOxidation.csv")
t <- X$Time..1000.min.
h <- X$h..in.A.
plot(t,h)
fit <- lm(h~t)
abline(fit$coefficients,type="line",col="red")

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  graphical parameter
"type" is obsolete

plot(fit$residuals)
qqnorm(fit$residuals)
fit

##
## Call:
## lm(formula = h ~ t)
##
## Coefficients:
## (Intercept)            t
```



```
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.

> getwd()
[1] "/home/guru/Desktop/DealingWithMaterialsData"
> X <- read.csv("Data/SiliconOxidation.csv")
> t <- X$Time..1000.min.
> h <- X$h..in.A.
> plot(t,h)
> fit <- lm(h~t)
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
>
```

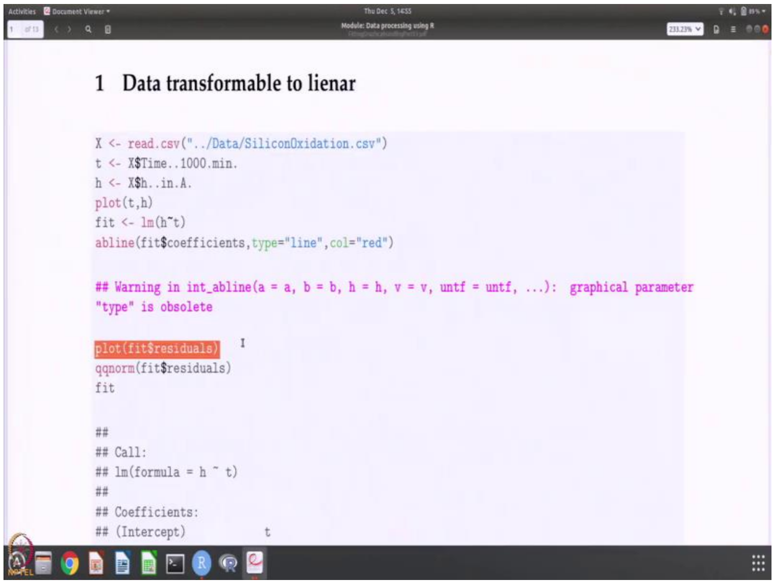
Data	
fit	List of 12
X	14 obs. of 2 variables
Values	
h	int [1:14] 161 184 174 199 258 274 3…
t	num [1:14] 0.95 0.96 1.44 1.44 1.92 …

So, let us do the exercise as usual, let us start R, we have version 3.6.1 and it is a good idea to know the working directory, so we are in the right directory and let us do the first exercise, so I am going to take this, so what is it that we are doing? We are reading the data and the data is for silicon oxidation and it is in csv format, so read csv is going to read it into the variable X and then we are going to store the time information in the variable small t and the thickness information in this variable small h and the first thing is to plot the time versus the thickness data and then try to fit it to a linear model h is related linearly to time.

And then draw a line for our fit and we are going to draw this line in red colour and it is a line and so we are going to have this data and we are going to have this fit and we are going to see how it looks, so let us do that. So, there are these data points and if you actually go to this paper, you will find that there are two sets of data for almost every temperature, because these are two different models using which they have calculated these values.

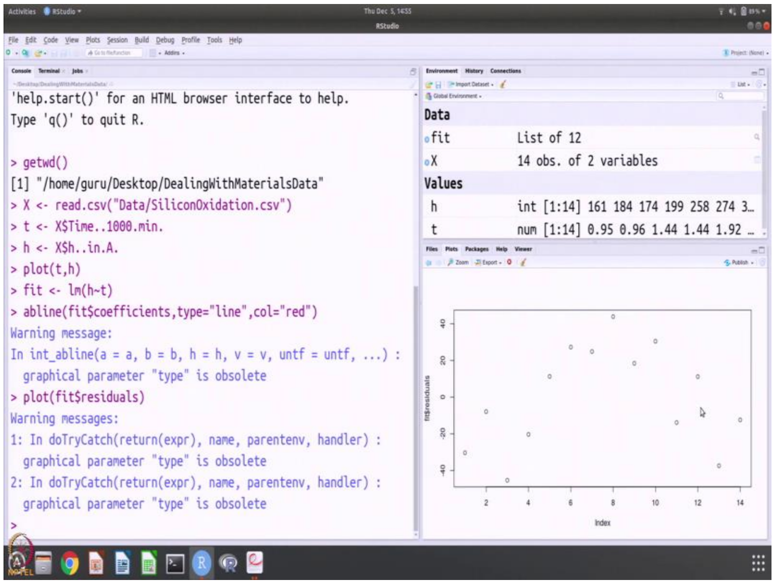
And this is the time, remember 1 means a 1000 minutes and h in angstrom, so it is 200 angstrom to about 500 angstroms is what we have got. And this is the fitted line and you can see that it is good because the data is close to many points and of course you can also do our usual exercise to know how good is our fit by looking at the residuals and plotting the residuals.

(Refer Slide Time: 10:28)



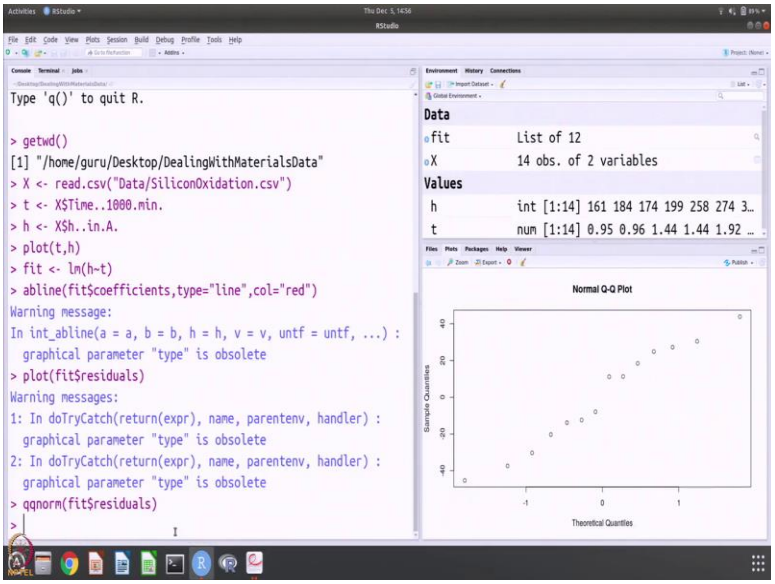
1 Data transformable to lienar

```
X <- read.csv("../Data/SiliconOxidation.csv")
t <- X$Time..1000.min.
h <- X$h..in.A.
plot(t,h)
fit <- lm(h~t)
abline(fit$coefficients,type="line",col="red")

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  graphical parameter
"type" is obsolete

plot(fit$residuals)
qqnorm(fit$residuals)
fit


##
## Call:
## lm(formula = h ~ t)
##
## Coefficients:
## (Intercept)            t
```



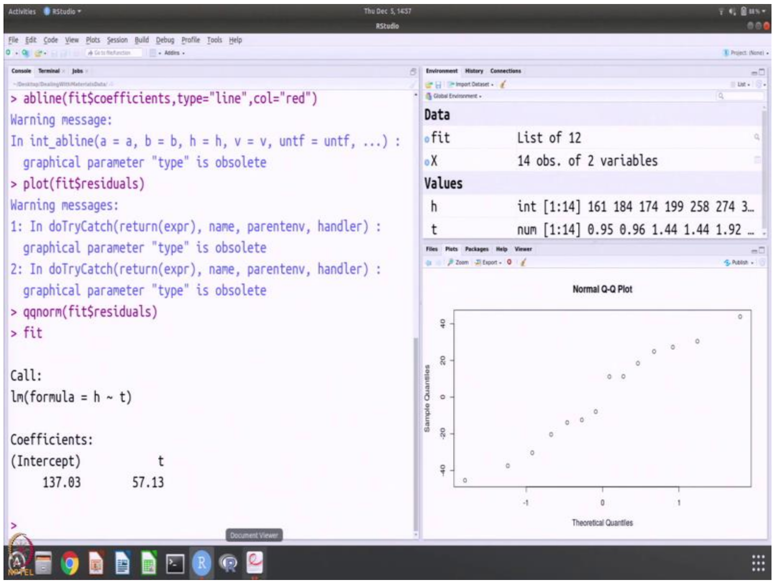
```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> getwd()
[1] "/home/guru/Desktop/DealingWithMaterialsData"
> X <- read.csv("Data/SiliconOxidation.csv")
> t <- X$Time..1000.min.
> h <- X$h..in.A.
> plot(t,h)
> fit <- lm(h~t)
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
> plot(fit$residuals)
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
>
```

```
Data
fit     List of 12
X       14 obs. of 2 variables
Values
h       int [1:14] 161 184 174 199 258 274 3…
t       num [1:14] 0.95 0.96 1.44 1.44 1.92 …
```



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

> getwd()
[1] "/home/guru/Desktop/DealingWithMaterialsData"
> X <- read.csv("Data/SiliconOxidation.csv")
> t <- X$Time..1000.min.
> h <- X$h..in.A.
> plot(t,h)
> fit <- lm(h~t)
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
> plot(fit$residuals)
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
> qqnorm(fit$residuals)
>
```

```
Data
fit     List of 12
X       14 obs. of 2 variables
Values
h       int [1:14] 161 184 174 199 258 274 3…
t       num [1:14] 0.95 0.96 1.44 1.44 1.92 …
```



```
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
> plot(fit$residuals)
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
> qqnorm(fit$residuals)
> fit

Call:
lm(formula = h ~ t)

Coefficients:
(Intercept)            t
     137.03        57.13

>
```

```
Data
fit        List of 12
X          14 obs. of 2 variables
Values
h          int [1:14] 161 184 174 199 258 274 3…
t          num [1:14] 0.95 0.96 1.44 1.44 1.92 …
```

So, you can plot the residuals and you see that it is, it is not quite random, there are data points both below 0 and above 0, but they have sort of relationships at these two end, it is below 0 and in between it is above 0, but you will see that this is not very uncommon and today we are going to see in this session several and in the next session several such data, which shows this kind of behaviour.

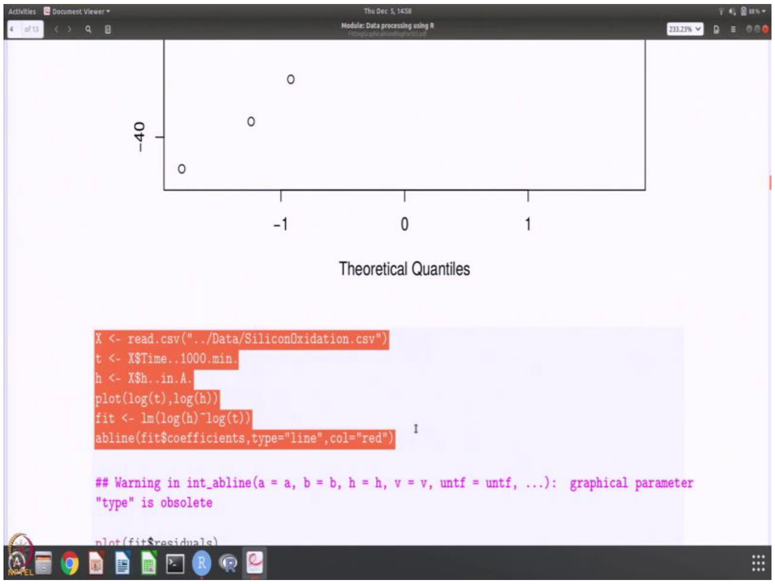
So, you can do a qqnorm, so is this a straight line may be quite close to a straight line. So, one might think that, so this fits it well, but we know that so let us look at what is our fit tell us. So, the fit gives an intercept of 137.03, what is the meaning of intercept? It is saying that there is a thickness of 137 when the time is 0. It is quite possible that you started with some oxide layer and that was some 137 angstroms and so as time goes by, because your starting point is at 137 angstroms you will get 137 angstroms.

But if that is not the case, then fitting this data to this straight line is not quite right, because at time t equal to 0 suppose you had silicon without any oxidation then the fit should actually give you 0 at 0 time and then slowly it should build it up. So, it crucially depends on whether the data actually was such that you had 137 angstroms to begin with in which case it is meaningful, so the fit is okay and we are doing fine. If that was not the case then the fit is not correct and we should not be using this.

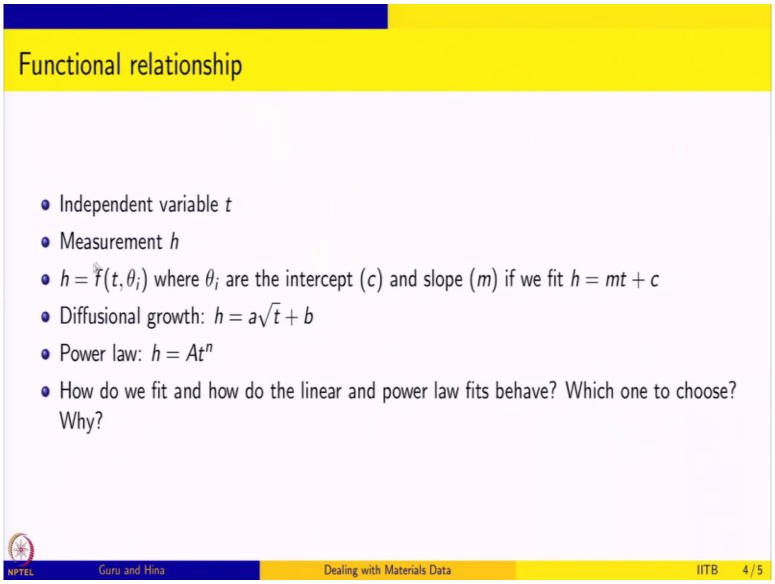
And we know that because this is oxidation of silicon and that is a diffusion driven process, it should have t to the power minus, t to the power half or some t to the power n kind of relationship?

So, it is not correct that if you just extend when you have 0 minutes it will cut at some value of h, which happens to be a 137 in this case, which is not quite right.

(Refer Slide Time: 13:01)

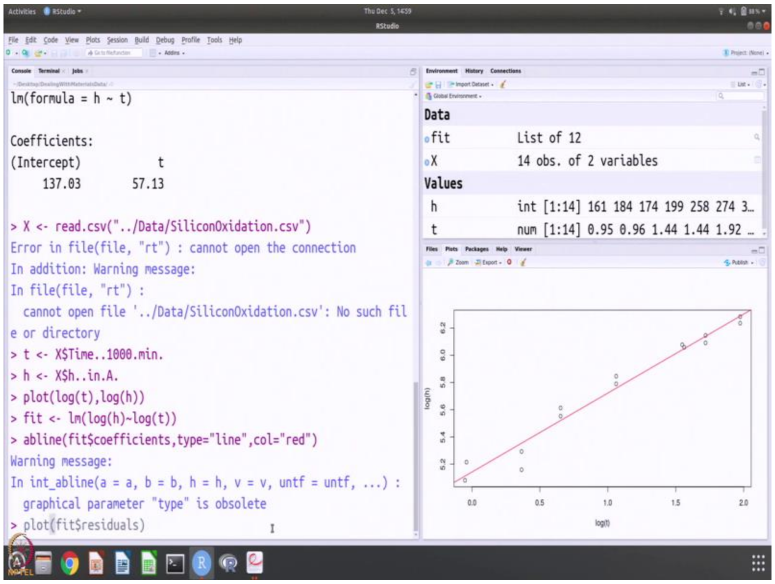


```
X <- read.csv("../Data/SiliconOxidation.csv")
t <- X$Time..1000.min.
h <- X$h..in.A.
plot(log(t),log(h))
fit <- lm(log(h)~log(t))
abline(fit$coefficients,type="line",col="red")

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  graphical parameter
"type" is obsolete

plot(fit$residuals)
```



Functional relationship

- Independent variable t
- Measurement h
- $h = f(t, \theta_i)$ where θ_i are the intercept (c) and slope (m) if we fit $h = mt + c$
- Diffusional growth: $h = a\sqrt{t} + b$
- Power law: $h = At^n$
- How do we fit and how do the linear and power law fits behave? Which one to choose? Why?



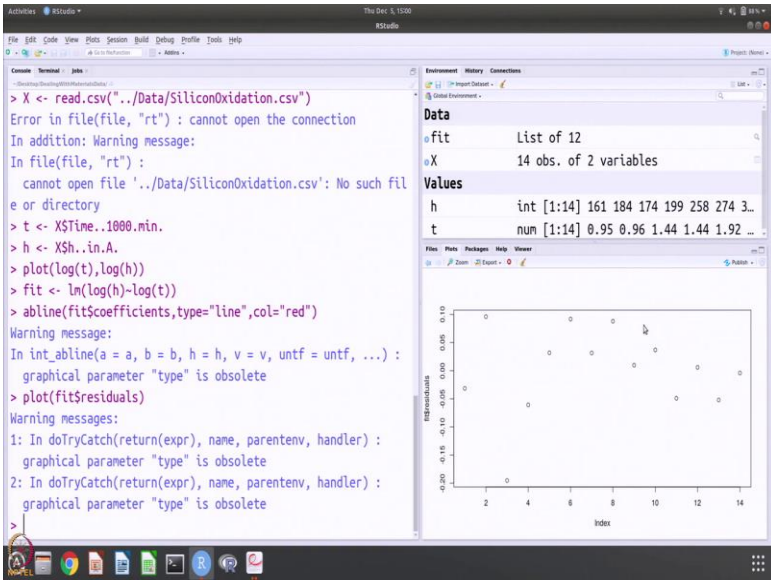
```
lm(formula = h ~ t)

Coefficients:
(Intercept)            t
     137.03        57.13

> X <- read.csv("../Data/SiliconOxidation.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file '../Data/SiliconOxidation.csv': No such fil
e or directory
> t <- X$Time..1000.min.
> h <- X$h..in.A.
> plot(log(t),log(h))
> fit <- lm(log(h)~log(t))
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
> plot(fit$residuals)
```

Data

fit	List of 12
X	14 obs. of 2 variables

Values

h	int [1:14] 161 184 174 199 258 274 3…
t	num [1:14] 0.95 0.96 1.44 1.44 1.92 …



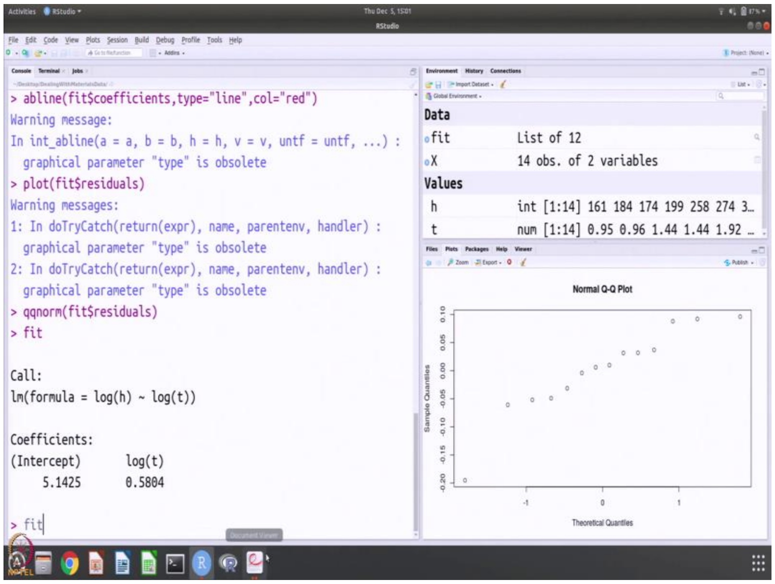
```
> X <- read.csv("../Data/SiliconOxidation.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file '../Data/SiliconOxidation.csv': No such fil
e or directory
> t <- X$Time..1000.min.
> h <- X$h..in.A.
> plot(log(t),log(h))
> fit <- lm(log(h)~log(t))
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
> plot(fit$residuals)
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
>
```

Data

fit	List of 12
X	14 obs. of 2 variables

Values

h	int [1:14] 161 184 174 199 258 274 3…
t	num [1:14] 0.95 0.96 1.44 1.44 1.92 …



```
> abline(fit$coefficients,type="line",col="red")
Warning message:
In int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  graphical parameter "type" is obsolete
> plot(fit$residuals)
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  graphical parameter "type" is obsolete
> qqnorm(fit$residuals)
> fit

Call:
lm(formula = log(h) ~ log(t))

Coefficients:
(Intercept)       log(t)
     5.1425       0.5804

> fit
```

```
Data
fit     List of 12
X       14 obs. of 2 variables
Values
h       int [1:14] 161 184 174 199 258 274 3…
t       num [1:14] 0.95 0.96 1.44 1.44 1.92 …
```

So, we want to now try and explore the other way, so let us do this exercises. And what is it that we are doing in this case? So, we have the same data and the same time and the thickness we have marked, but now we are going to plot logarithm of time with logarithm of thickness, as you can see, this is something that we discussed earlier

So, if you have logarithms, so if you take logarithms and you can see that logarithm of the oxidation the thickness times logarithm of this quantity is going to give you some if you have logarithm of t if you take, then t to the power half if it is fitting then it should give you some factor of half coming in the front.

And so this will become some constant in the logarithmic scale. Similarly, if it is power law, so this is what we are trying to fit, logarithm of h to logarithm of t will give you a linear fit and because it is logarithm of t the n will come to the front and so you will just have it as a constant coming up, so that is what we are trying to do here.

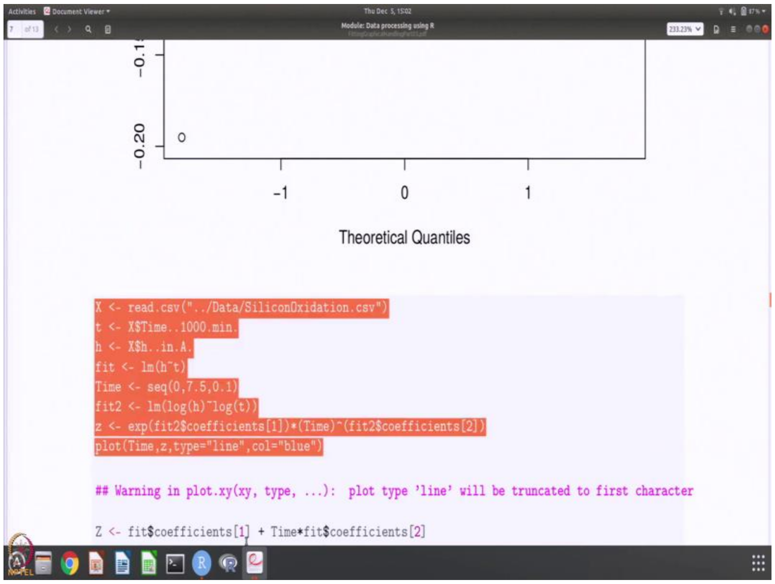
So, let us do this exercise, so I am going to take log t and log h, I am going to plot it, I am also going to fit log h to log t and then we are going to do the same thing, so take the fitted line and plot it in red as a line and you have the data points. So, you can see that log t versus log h actually gives you also a straight line, this is also quite close to the points that you have.

And you can also look at the residual, so again residuals are spread out in this case there is not quite symmetric spread lots of data points are quite close to 0 on either side and this is sort of out layer, so most of the data falls between minus 0.05 and 0.1 and of course, there are some data on the positive side, some data on the negative side and they are not quite equally spread but it is not too bad either.

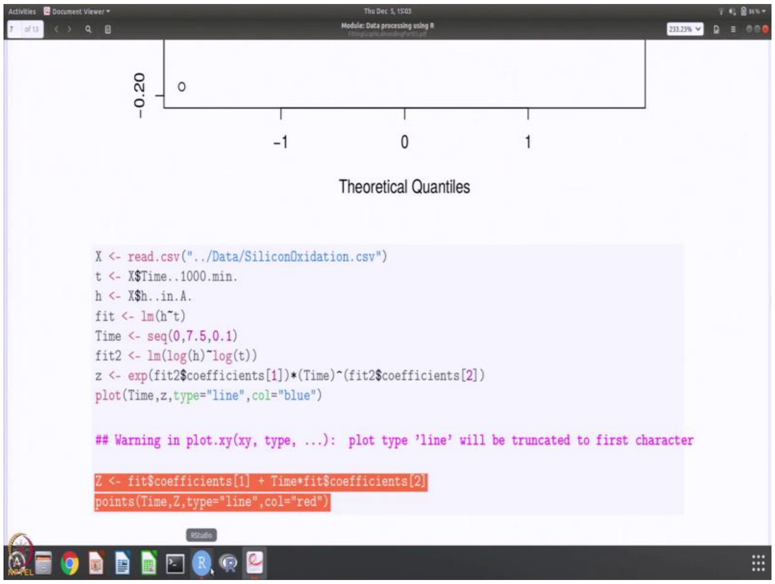
And you can of course do the qqnorm and you can see that this is really not a straight line. So, the data should be like that, but that is not what we see, so if, the error does not seem to be normally distributed. So, that is what it means, but we know from the problem that it should have a relationship of this kind of course you can just give the fit and then look at the numbers.

So, if you look at the fit, it actually gives you a coefficient of 5.1425 for intercept and because this is log h versus log t and if you try to calculate what happens to this parameter, you will find some nonzero number for that also. So, is that correct? I mean are we doing the right thing is the question, so this is how it goes. And log h versus log t you can see when it is 0 in the logarithm scale at least it give some numbers here. So, let us explore it a little bit further try to understand what is happing, so we have already seen these residuals and qq plots.

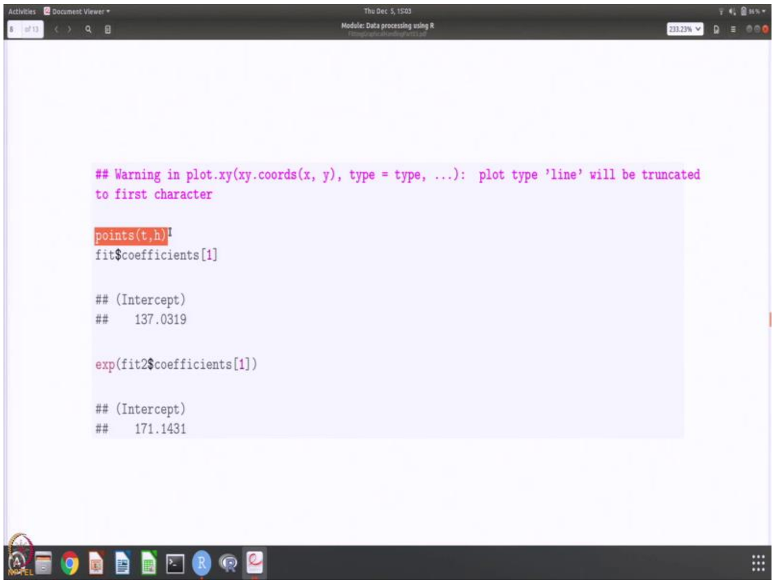
(Refer Slide Time: 17:01)



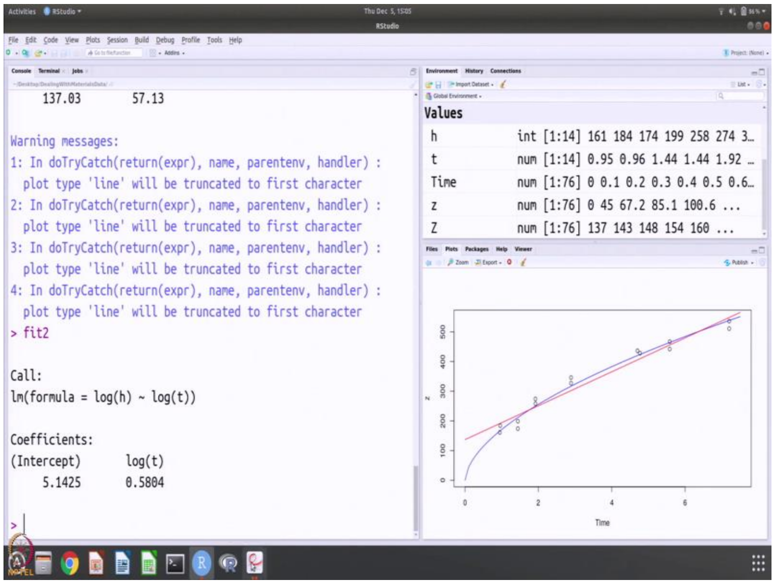
```
X <- read.csv("../Data/SiliconOxidation.csv")
t <- X$Time..1000.min.
h <- X$h..in.A.
fit <- lm(h~t)
Time <- seq(0,7.5,0.1)
fit2 <- lm(log(h)~log(t))
z <- exp(fit2$coefficients[1])*(Time)^(fit2$coefficients[2])
plot(Time,z,type="line",col="blue")

## Warning in plot.xy(xy, type, ...):  plot type 'line' will be truncated to first character

Z <- fit$coefficients[1] + Time*fit$coefficients[2]
```



```
X <- read.csv("../Data/SiliconOxidation.csv")
t <- X$Time..1000.min.
h <- X$h..in.A.
fit <- lm(h~t)
Time <- seq(0,7.5,0.1)
fit2 <- lm(log(h)~log(t))
z <- exp(fit2$coefficients[1])*(Time)^(fit2$coefficients[2])
plot(Time,z,type="line",col="blue")

## Warning in plot.xy(xy, type, ...):  plot type 'line' will be truncated to first character

Z <- fit$coefficients[1] + Time*fit$coefficients[2]
points(Time,Z,type="line",col="red")
```



```
## Warning in plot.xy(xy.coords(x, y), type = type, ...):  plot type 'line' will be truncated
to first character

points(t,h)
fit$coefficients[1]

## (Intercept)
##    137.0319

exp(fit2$coefficients[1])

## (Intercept)
##    171.1431
```



```
    137.03       57.13

Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  plot type 'line' will be truncated to first character
2: In doTryCatch(return(expr), name, parentenv, handler) :
  plot type 'line' will be truncated to first character
3: In doTryCatch(return(expr), name, parentenv, handler) :
  plot type 'line' will be truncated to first character
4: In doTryCatch(return(expr), name, parentenv, handler) :
  plot type 'line' will be truncated to first character
> fit2

Call:
lm(formula = log(h) ~ log(t))

Coefficients:
(Intercept)       log(t)
     5.1425       0.5804

>
```

```
Values
h          int [1:14] 161 184 174 199 258 274 3…
t          num [1:14] 0.95 0.96 1.44 1.44 1.92 …
Time       num [1:76] 0 0.1 0.2 0.3 0.4 0.5 0.6…
z          num [1:76] 0 45 67.2 85.1 100.6 ...
Z          num [1:76] 137 143 148 154 160 ...
```

Now, I am going to take the data, so we are going to take the data, we are going do a linear fit and then I am going to do a second fit which is logarithmic and using the fitted coefficients I am going to calculate the value for the so I am going to take because time goes from 0 to some 7000 minutes, so I am going to go 0 to 7.5 in 1000 of minutes.

So, 7500 is what I am going to make a sequence of time and for that I am going to fit these parameters. And then I am going to plot the fit which is 2 logarithm h to logarithmic t, so that we are going to plot first, So, you can see that in this case where time is 0 the thickness is also 0, so which is a good sign so we are having the right sort of behaviour

And let us plot the other fit the linear fit on the same plot, so you have this as the linear plot and this blue line as the power law plot and let us put the data points on the same curve, so know, to know how they look like, so here are the data points. So, if you just look at the data points both the red line and the blue line seem to be okay.

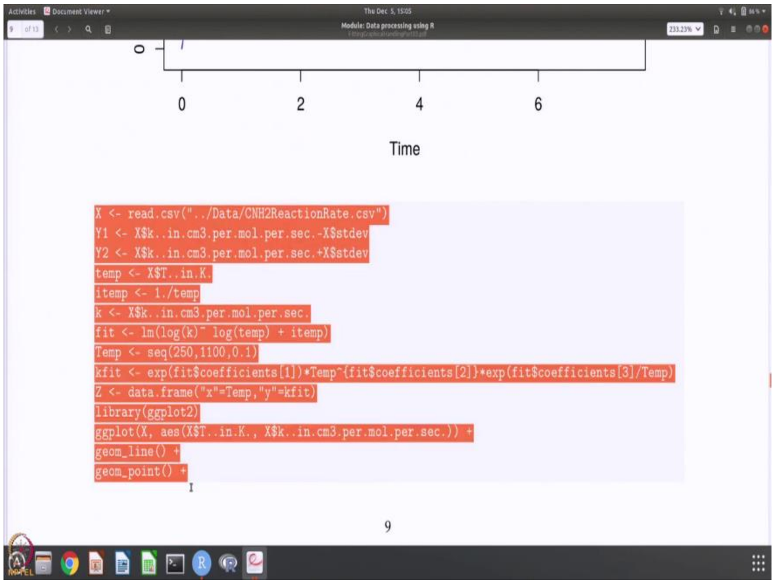
And like I said, if we started at time 0 with the some oxide thickness, then this curve makes sense, so it is not that it is wrong under certain conditions that this could be the right curve, but if we have the time at time 0 the oxide thickness to be 0 then this is the right kind of curve that we should fit.

So, it will fit for all time properly and as we can see as the time increases this will deviate so straight line goes like this and this one goes like this. So, there will be deviations at this end and there will be deviations outside of this end, so if they are important then it is important to fit to the right form.

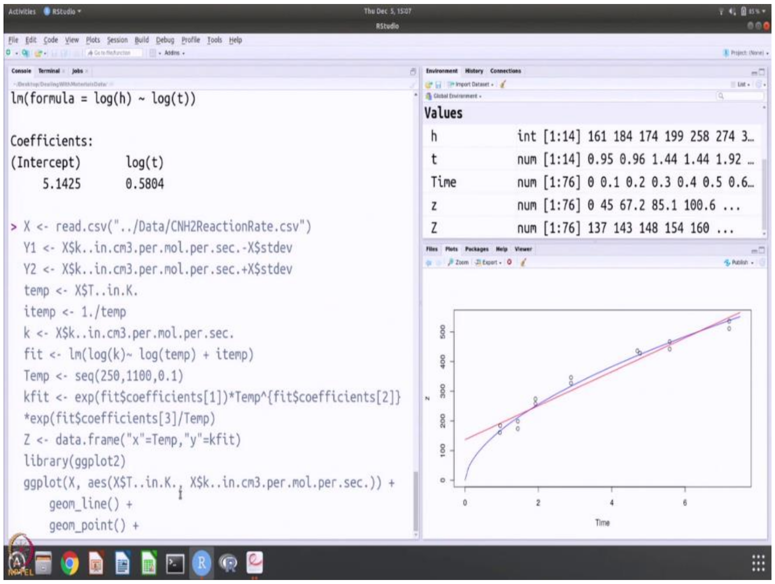
And I also want you to pay attention to this fact that if I just look at the fit parameters, fit and fit[2] just by looking at the parameters, you cannot make out that this actually gives 0 intercept at t equal to 0, that is because you have to actually substitute in that expression and it might give some number but something else that multiplies it is when t is 0, 0 to the power something so that will give you actually a very small number and make it 0.

So, if you just look at the intercept you might not even realize, that is why it is always a good idea to plot and look at the plot, after you do the fitting, you should always plot the fitted line along with the data and analyze it. So, it is very very important and here is an example where just by looking at the fit coefficients you would not have realized that actually this line is going towards 0. But if you do the algebra and if you do the fitting you can see that it does. So, this is the first exercise, this is for oxidation of silicon.

(Refer Slide Time: 20:55)



```
X <- read.csv("../Data/CNH2ReactionRate.csv")
Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
temp <- X$T..in.K.
itemp <- 1./temp
k <- X$k..in.cm3.per.mol.per.sec.
fit <- lm(log(k)~ log(temp) + itemp)
Temp <- seq(250,1100,0.1)
kfit <- exp(fit$coefficients[1])*Temp^{fit$coefficients[2]}*exp(fit$coefficients[3]/Temp)
Z <- data.frame("x"=Temp,"y"=kfit)
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
geom_line() +
geom_point() +
```



```
lm(formula = log(h) ~ log(t))

Coefficients:
(Intercept)       log(t)
     5.1425       0.5804

> X <- read.csv("../Data/CNH2ReactionRate.csv")
  Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
  Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
  temp <- X$T..in.K.
  itemp <- 1./temp
  k <- X$k..in.cm3.per.mol.per.sec.
  fit <- lm(log(k)~ log(temp) + itemp)
  Temp <- seq(250,1100,0.1)
  kfit <- exp(fit$coefficients[1])*Temp^{fit$coefficients[2]}
  *exp(fit$coefficients[3]/Temp)
  Z <- data.frame("x"=Temp,"y"=kfit)
  library(ggplot2)
  ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
      geom_line() +
      geom_point() +
```

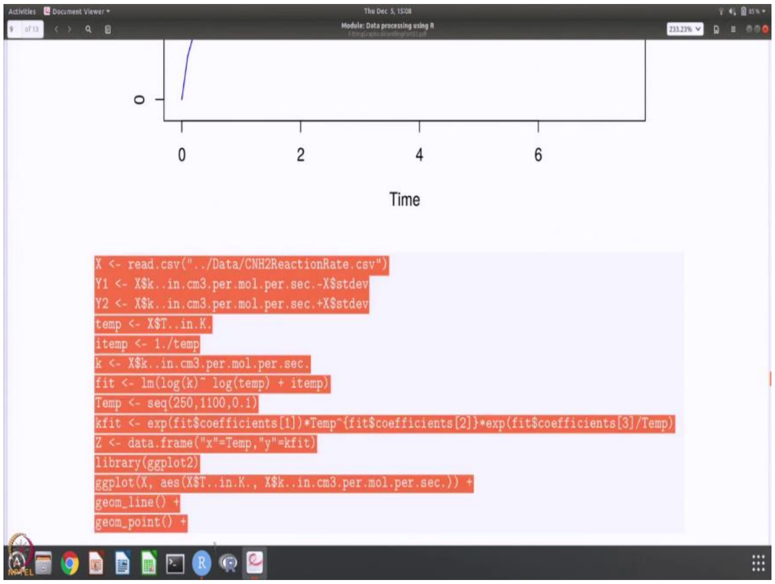
```
Values
h       int [1:14] 161 184 174 199 258 274 3…
t       num [1:14] 0.95 0.96 1.44 1.44 1.92 …
Time    num [1:76] 0 0.1 0.2 0.3 0.4 0.5 0.6…
z       num [1:76] 0 45 67.2 85.1 100.6 ...
Z       num [1:76] 137 143 148 154 160 ...
```

Let us do our second exercise and in this exercise, so we are looking at the reaction rate data, so let me do this, so we read the data and we store the standard deviation is given, so the y-min and y-max, so this is basically the spread of the data points, and if you look at the data, you see that the standard deviation is not the same as temperature increases the error also increases and then there is temperature and this is inverse of temperature, so I have called it itemp and k is actually the reaction rate.

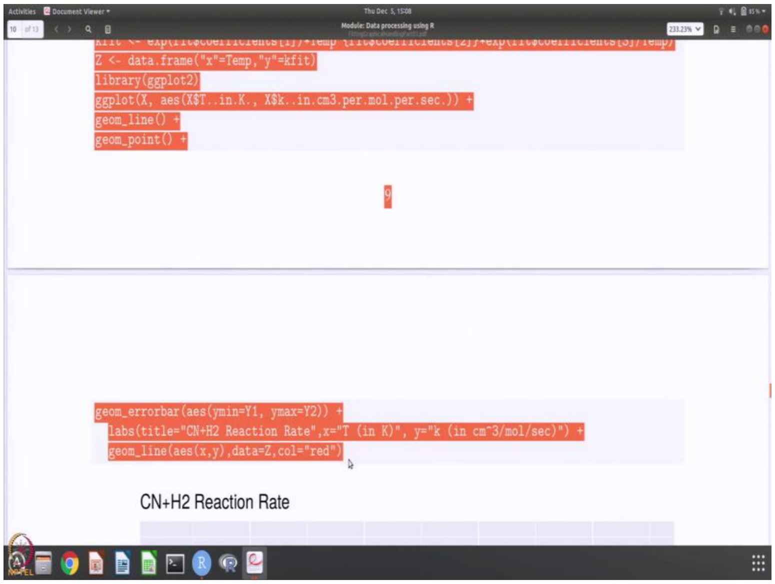
So, what we are trying to do is to fit logarithm k to logarithm temperature plus inverse of temperature, because remember we said that it is A t to the power some B and exponential minus EA by RT. So, this 1 by t relationship is here and when you take log so the B will come to the front. So, it is B log t and the constant term, so that is what we are trying to fit.

And after we fit we actually generate the sequence and for fitted parameters we calculate this, because exponential fit coefficient 1 is a constant, times temperature to the fit coefficient 2 and multiplied by exponential fit coefficient 3 by temperature, so this is the data that is from the fitting and so we plotted and of course after we plot the data from the fit we should also plot the line and the data should be plotted with the error bars, so that is the complete thing that we want to do.

(Refer Slide Time: 22:30)

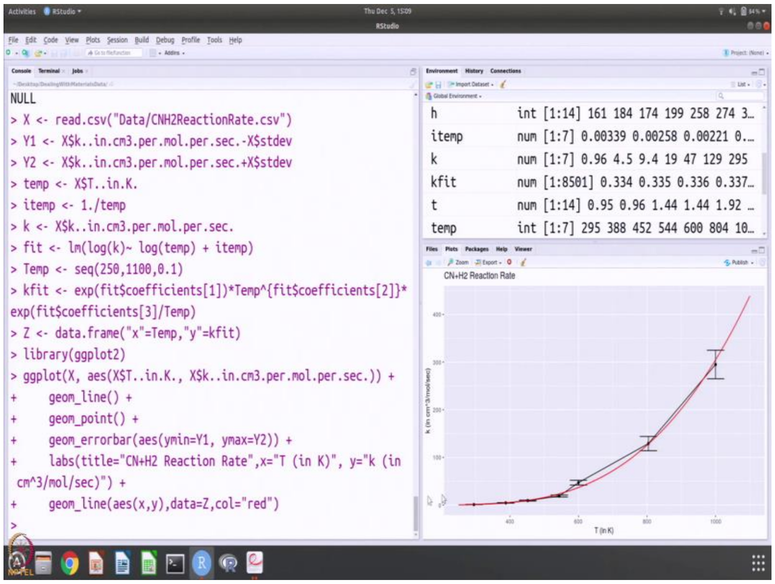


```
X <- read.csv("../Data/CNH2ReactionRate.csv")
Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
temp <- X$T..in.K.
itemp <- 1./temp
k <- X$k..in.cm3.per.mol.per.sec.
fit <- lm(log(k)~ log(temp) + itemp)
Temp <- seq(250,1100,0.1)
kfit <- exp(fit$coefficients[1])*Temp^{fit$coefficients[2]}*exp(fit$coefficients[3]/Temp)
Z <- data.frame("x"=Temp,"y"=kfit)
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
geom_line() +
geom_point() +
```



```
Z <- data.frame("x"=Temp,"y"=kfit)
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
geom_line() +
geom_point() +
```

```
geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
  labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in cm^3/mol/sec)") +
  geom_line(aes(x,y),data=Z,col="red")
```

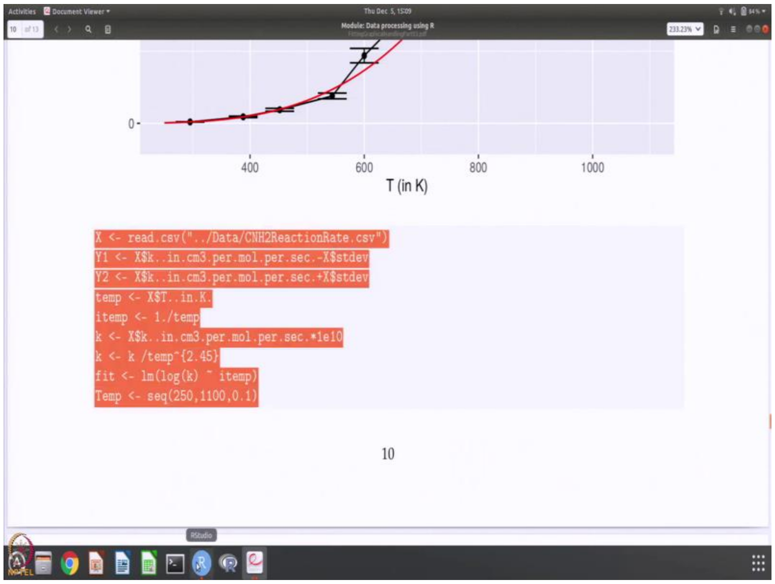


```
NULL
> X <- read.csv("Data/CNH2ReactionRate.csv")
> Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
> Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
> temp <- X$T..in.K.
> itemp <- 1./temp
> k <- X$k..in.cm3.per.mol.per.sec.
> fit <- lm(log(k)~ log(temp) + itemp)
> Temp <- seq(250,1100,0.1)
> kfit <- exp(fit$coefficients[1])*Temp^{fit$coefficients[2]}*
exp(fit$coefficients[3]/Temp)
> Z <- data.frame("x"=Temp,"y"=kfit)
> library(ggplot2)
> ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
+     geom_line() +
+     geom_point() +
+     geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
+     labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in
 cm^3/mol/sec)") +
+     geom_line(aes(x,y),data=Z,col="red")
>
```

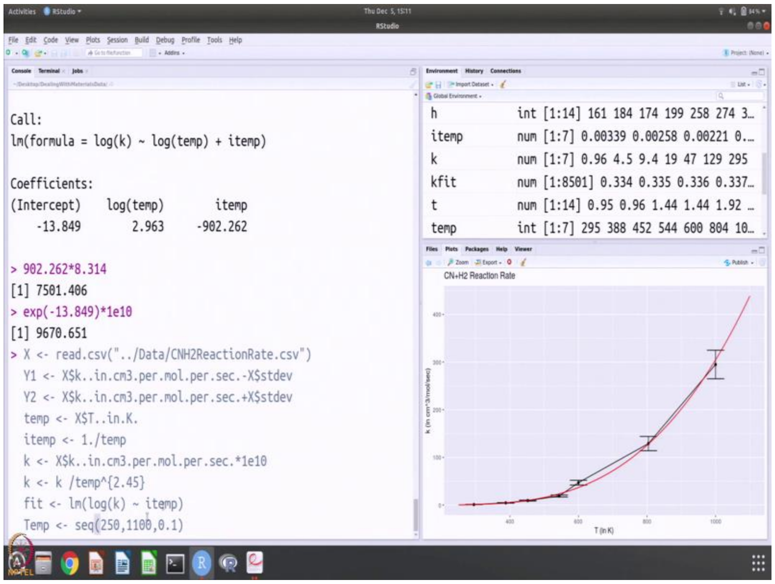
```
h         int [1:14] 161 184 174 199 258 274 3…
itemp     num [1:7] 0.00339 0.00258 0.00221 0.…
k         num [1:7] 0.96 4.5 9.4 19 47 129 295
kfit      num [1:8501] 0.334 0.335 0.336 0.337…
t         num [1:14] 0.95 0.96 1.44 1.44 1.92 …
temp      int [1:7] 295 388 452 544 600 804 10…
```

So, there is something let us redo this, so we want to do this exercise, so you can see that we have plotted the data and as you go to higher and higher temperatures, the error bars are increasing and this is the black is the data and then we have drawn the red line, which is from our fitting, with data Z, Z was obtained by fitting and that is the line that goes through. So, this is the first exercise of course.

(Refer Slide Time: 24:05)



```
X <- read.csv("../Data/CNH2ReactionRate.csv")
Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
temp <- X$T..in.K.
itemp <- 1./temp
k <- X$k..in.cm3.per.mol.per.sec.*1e10
k <- k /temp^{2.45}
fit <- lm(log(k) ~ itemp)
Temp <- seq(250,1100,0.1)
```



```
Call:
lm(formula = log(k) ~ log(temp) + itemp)

Coefficients:
(Intercept)    log(temp)        itemp
    -13.849        2.963     -902.262

> 902.262*8.314
[1] 7501.406
> exp(-13.849)*1e10
[1] 9670.651
> X <- read.csv("../Data/CNH2ReactionRate.csv")
  Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
  Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
  temp <- X$T..in.K.
  itemp <- 1./temp
  k <- X$k..in.cm3.per.mol.per.sec.*1e10
  k <- k /temp^{2.45}
  fit <- lm(log(k) ~ itemp)
  Temp <- seq(250,1100,0.1)
```

```
h        int [1:14] 161 184 174 199 258 274 3…
itemp    num [1:7] 0.00339 0.00258 0.00221 0.…
k        num [1:7] 0.96 4.5 9.4 19 47 129 295
kfit     num [1:8501] 0.334 0.335 0.336 0.337…
t        num [1:14] 0.95 0.96 1.44 1.44 1.92 …
temp     int [1:7] 295 388 452 544 600 804 10…
```

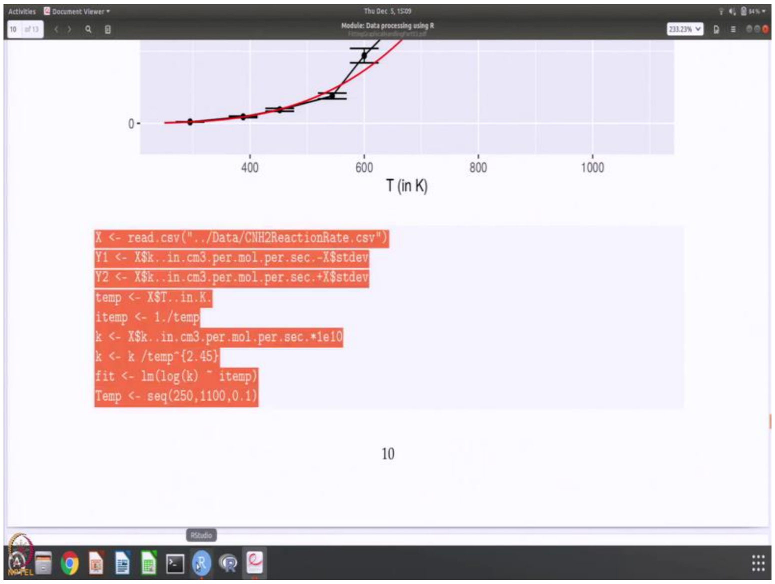
And if you look at the numbers and if you compare it with the paper you realize that they are not matching that is because the k is actually given in 10 to the power 10 centimetre cube per mole per second, so let us do that exercise. So, all that we are going to do is that we will correct for it, here we did not do that, so in this case we are going to correct for it and we are also going to plot only the logarithm of k to the temperature because I am going to assume in this case that it is 2.45, because when you do the generic fitting you do not get that parameters.

So, that you can check you can look at the fit parameter, so you see that the logarithm of temperature has a coefficient 2.9, it is not 2.45, but there are reasons to believe that it is 2.45 in which case we have to explicitly make it 2.45 and then fit and see how these values. Because of this if you see this is 902, so 902.262 into 8.314 is actually the activation energy. So, it gives you something like 7.5 kilo joules, but the fitting actually gives 9.

And this exponential minus 13.849 multiplied by 1 e power 10, so this gives you 96 but the paper gives you a fitting which is like order of 10 to the power 5. So, these differences are coming because the, it seems to be fitting this to some exponent which is not the right exponent. Theoretically we know it should be 2.45, so the next exercise is actually to make sure that we fit it for that 2.45 and that is what we are doing here.

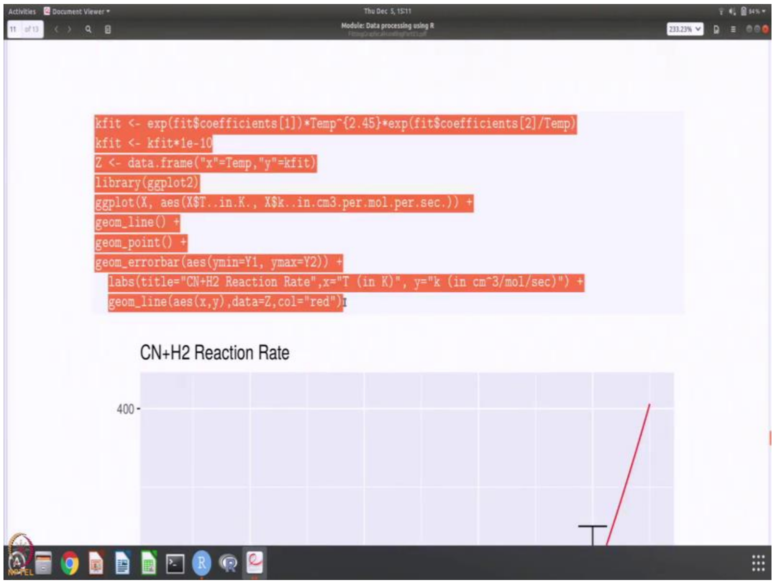
So, we take, we account for this 10 to the power 10 and we also divide the k by t to the power 2.45 and that is the quantity that we are fitting. So, we have k by t to the power 2.45 which is equal to some constant times exponential minus EA by RT, so you can take logarithm of this quantity and fit it against 1 by t and find out how the fitting works.

(Refer Slide Time: 26:33)

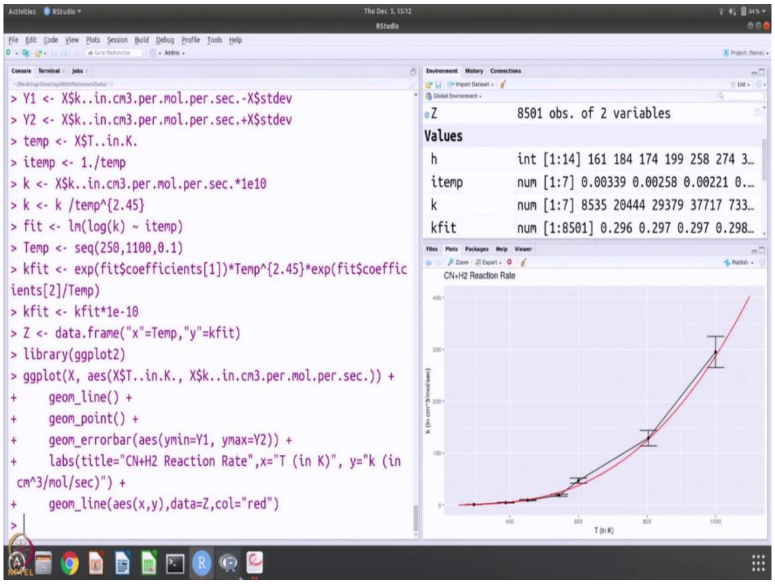


```
X <- read.csv("../Data/CNH2ReactionRate.csv")
Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
temp <- X$T..in.K.
itemp <- 1./temp
k <- X$k..in.cm3.per.mol.per.sec.*1e10
k <- k /temp^{2.45}
fit <- lm(log(k) ~ itemp)
Temp <- seq(250,1100,0.1)
```

10

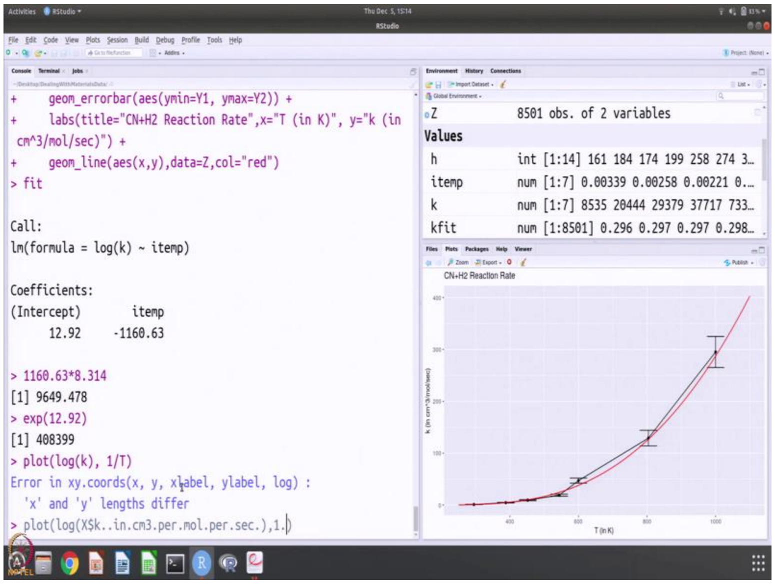


```
kfit <- exp(fit$coefficients[1])*Temp^{2.45}*exp(fit$coefficients[2]/Temp)
kfit <- kfit*1e-10
Z <- data.frame("x"=Temp,"y"=kfit)
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
geom_line() +
geom_point() +
geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
  labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in cm^3/mol/sec)") +
  geom_line(aes(x,y),data=Z,col="red")
```



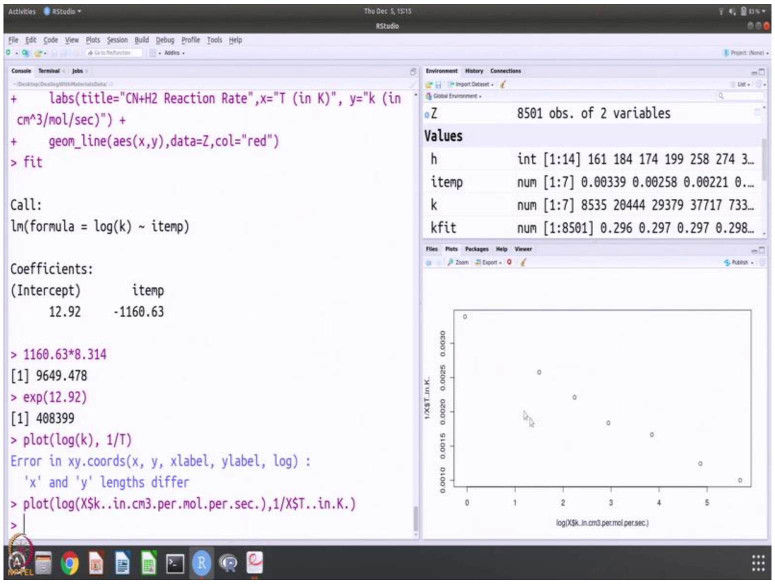
```
> Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
> Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
> temp <- X$T..in.K.
> itemp <- 1./temp
> k <- X$k..in.cm3.per.mol.per.sec.*1e10
> k <- k /temp^{2.45}
> fit <- lm(log(k) ~ itemp)
> Temp <- seq(250,1100,0.1)
> kfit <- exp(fit$coefficients[1])*Temp^{2.45}*exp(fit$coeffic
ients[2]/Temp)
> kfit <- kfit*1e-10
> Z <- data.frame("x"=Temp,"y"=kfit)
> library(ggplot2)
> ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
+     geom_line() +
+     geom_point() +
+     geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
+     labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in
 cm^3/mol/sec)") +
+     geom_line(aes(x,y),data=Z,col="red")
> 
```

```
Z       8501 obs. of 2 variables
Values
h       int [1:14] 161 184 174 199 258 274 3…
itemp   num [1:7] 0.00339 0.00258 0.00221 0.…
k       num [1:7] 8535 20444 29379 37717 733…
kfit    num [1:8501] 0.296 0.297 0.297 0.298…
```



```
+     geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
+     labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in
 cm^3/mol/sec)") +
+     geom_line(aes(x,y),data=Z,col="red")
> fit

Call:
lm(formula = log(k) ~ itemp)

Coefficients:
(Intercept)        itemp
      12.92     -1160.63

> 1160.63*8.314
[1] 9649.478
> exp(12.92)
[1] 408399
> plot(log(k), 1/T)
Error in xy.coords(x, y, xlabel, ylabel, log) :
  'x' and 'y' lengths differ
> plot(log(X$k..in.cm3.per.mol.per.sec.),1.)
```

```
Z       8501 obs. of 2 variables
Values
h       int [1:14] 161 184 174 199 258 274 3…
itemp   num [1:7] 0.00339 0.00258 0.00221 0.…
k       num [1:7] 8535 20444 29379 37717 733…
kfit    num [1:8501] 0.296 0.297 0.297 0.298…
```



```
+     labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in
 cm^3/mol/sec)") +
+     geom_line(aes(x,y),data=Z,col="red")
> fit

Call:
lm(formula = log(k) ~ itemp)

Coefficients:
(Intercept)        itemp
      12.92     -1160.63

> 1160.63*8.314
[1] 9649.478
> exp(12.92)
[1] 408399
> plot(log(k), 1/T)
Error in xy.coords(x, y, xlabel, ylabel, log) :
  'x' and 'y' lengths differ
> plot(log(X$k..in.cm3.per.mol.per.sec.),1/X$T..in.K.)
> 
```

```
Z        8501 obs. of 2 variables
Values
h        int [1:14] 161 184 174 199 258 274 3…
itemp    num [1:7] 0.00339 0.00258 0.00221 0.…
k        num [1:7] 8535 20444 29379 37717 733…
kfit     num [1:8501] 0.296 0.297 0.297 0.298…
```

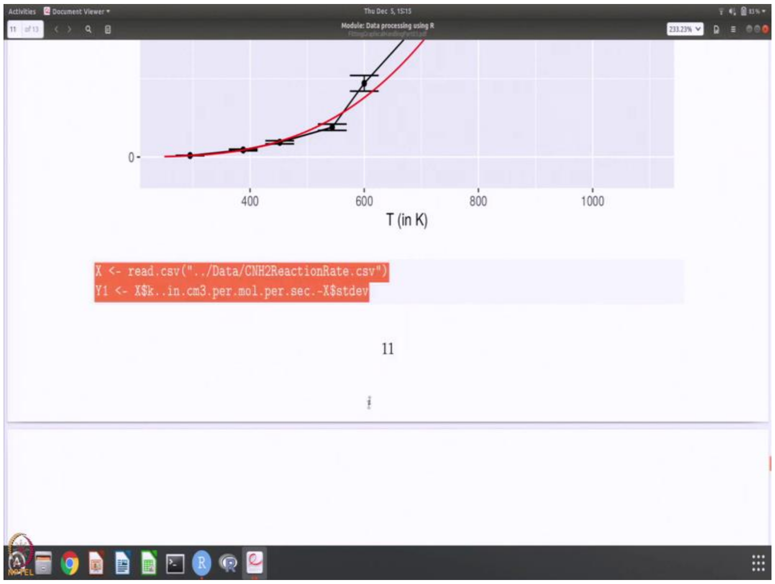
So, that is the exercise, let us do that and see how that works out, there is a problem, problem is because of some, so let me that is simply because of the way I cut pasted, so let me repeat that and make sure this time we do not have the error. So, you can see that now we have got a fit and in this case we have assumed that the exponential is 2.45 and we have got a fit.

So, that is what is shown and now if you calculate the fit parameters, so you can see 1160.63 and multiplied by 8.314, it actually gives you 9.6 kilo joule or so and this quantity 12.92 actually gives you a 4 into 10 to the power 5, if you look at the paper it is still gives this as some 3.1 into 10 to the power 5 and this as some 9 point something kilo joule.

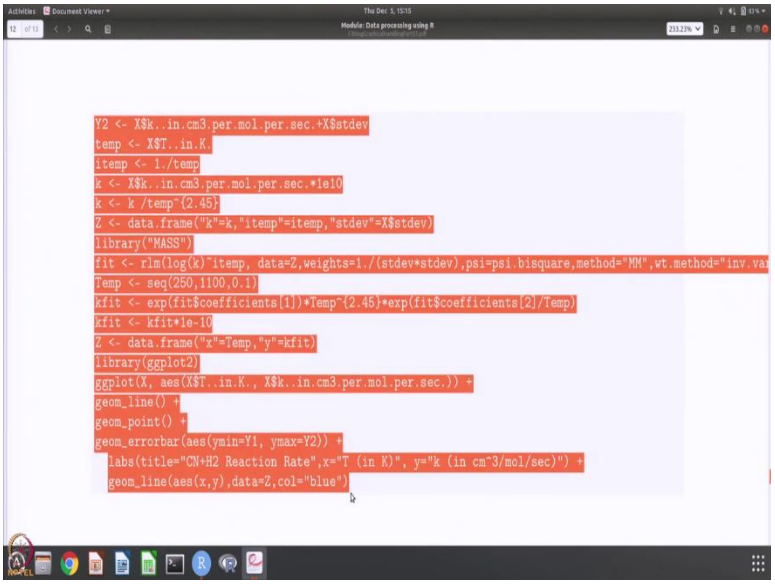
So, we are in the ballpark same figure, but there is slight difference between the numbers that is reported in the paper and what we are getting from our analysis. I believe this is because the analysis is done in their case by taking into account that different data points have different error values. So, when you have higher error, you want to give less weightage to those data points, compared to quantities, which have lower error, so you want them to have better fit for those points.

So, you can also see that this kind of plotting t versus k you see that these lower end data points you are not able to very clearly see, so if on the other hand if you plot logarithm of k versus 1 by T for example, I think 1 by temperature, so you can see this better and so this is because I mean it is the values are changing very much and at the lower end where they are clustered together, you can clearly see. So, this is one way of spreading the data out and seeing but of course logarithm also distorts at the other end, so one has to be careful when we are doing this.

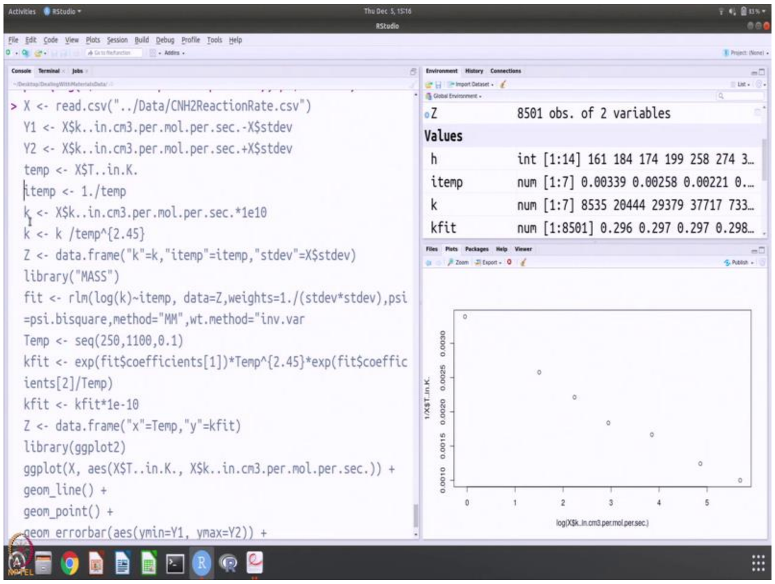
(Refer Slide Time: 30:12)



```
X <- read.csv("../Data/CNH2ReactionRate.csv")
Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
```



```
Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
temp <- X$T..in.K.
itemp <- 1./temp
k <- X$k..in.cm3.per.mol.per.sec.*1e10
k <- k /temp^{2.45}
Z <- data.frame("k"=k,"itemp"=itemp,"stdev"=X$stdev)
library("MASS")
fit <- rlm(log(k)~itemp, data=Z,weights=1./(stdev*stdev),psi=psi.bisquare,method="MM",wt.method="inv.va
Temp <- seq(250,1100,0.1)
kfit <- exp(fit$coefficients[1])*Temp^{2.45}*exp(fit$coefficients[2]/Temp)
kfit <- kfit*1e-10
Z <- data.frame("x"=Temp,"y"=kfit)
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
geom_line() +
geom_point() +
geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
  labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in cm^3/mol/sec)") +
  geom_line(aes(x,y),data=Z,col="blue")
```



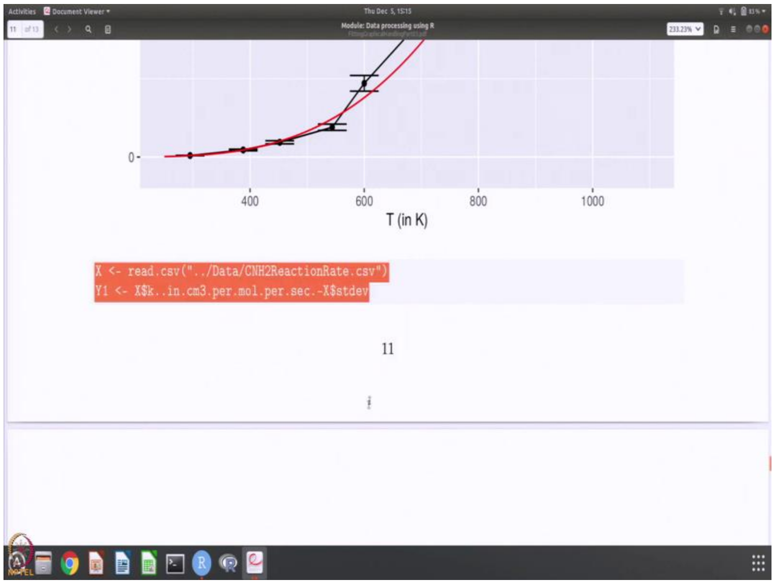
```
> X <- read.csv("../Data/CNH2ReactionRate.csv")
  Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
  Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
  temp <- X$T..in.K.
  itemp <- 1./temp
  k <- X$k..in.cm3.per.mol.per.sec.*1e10
  k <- k /temp^{2.45}
  Z <- data.frame("k"=k,"itemp"=itemp,"stdev"=X$stdev)
  library("MASS")
  fit <- rlm(log(k)~itemp, data=Z,weights=1./(stdev*stdev),psi
  =psi.bisquare,method="MM",wt.method="inv.var
  Temp <- seq(250,1100,0.1)
  kfit <- exp(fit$coefficients[1])*Temp^{2.45}*exp(fit$coeffic
  ients[2]/Temp)
  kfit <- kfit*1e-10
  Z <- data.frame("x"=Temp,"y"=kfit)
  library(ggplot2)
  ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
  geom_line() +
  geom_point() +
  geom errorbar(aes(ymin=Y1, ymax=Y2)) +
```

```
Z         8501 obs. of 2 variables
Values
h         int [1:14] 161 184 174 199 258 274 3…
itemp     num [1:7] 0.00339 0.00258 0.00221 0.…
k         num [1:7] 8535 20444 29379 37717 733…
kfit      num [1:8501] 0.296 0.297 0.297 0.298…
```

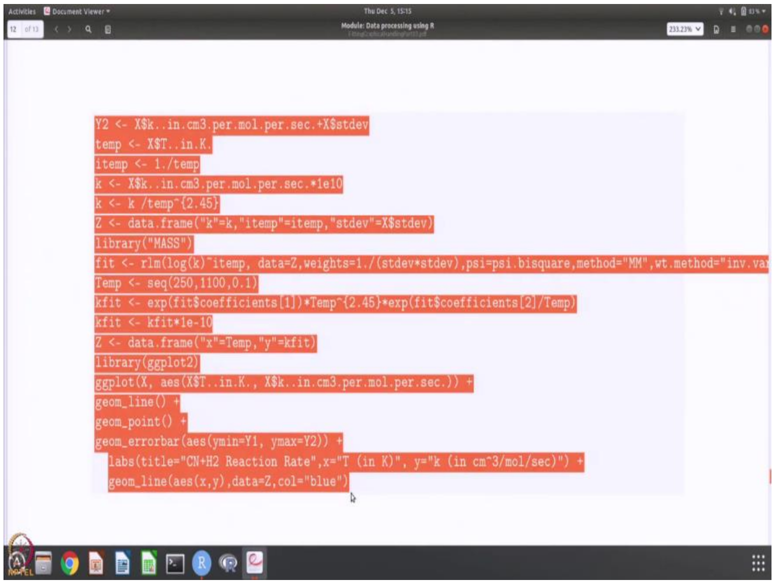
So, now the last exercise. I am going to show you how to take into account the fact that the different points have different statistical weights. So, let us take a look at what we are doing, so we read the data as usual, so we have the reaction rate data, so we have read the data and we have taken the values Y1 and Y2 and we are calling this as temperature, this is inverse of temperature and k is scaled by 10 to the power 10 and we are assuming that it is t to the power 2.45, so we are going to take this.

And we are going to use this library mass and then we are going to fit a robust linear model and for that we are going to give different weights and the weighting method is inverse variance.

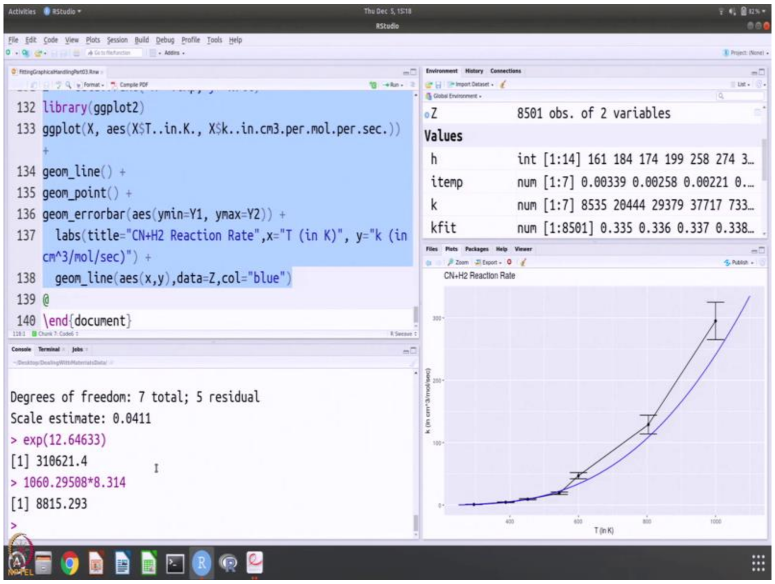
(Refer Slide Time: 32:03)



```
X <- read.csv("../Data/CNH2ReactionRate.csv")
Y1 <- X$k..in.cm3.per.mol.per.sec.-X$stdev
```



```
Y2 <- X$k..in.cm3.per.mol.per.sec.+X$stdev
temp <- X$T..in.K.
itemp <- 1./temp
k <- X$k..in.cm3.per.mol.per.sec.*1e10
k <- k /temp^{2.45}
Z <- data.frame("k"=k,"itemp"=itemp,"stdev"=X$stdev)
library("MASS")
fit <- rlm(log(k)~itemp, data=Z,weights=1./(stdev*stdev),psi=psi.bisquare,method="MM",wt.method="inv.va
Temp <- seq(250,1100,0.1)
kfit <- exp(fit$coefficients[1])*Temp^{2.45}*exp(fit$coefficients[2]/Temp)
kfit <- kfit*1e-10
Z <- data.frame("x"=Temp,"y"=kfit)
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.)) +
geom_line() +
geom_point() +
geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
  labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in cm^3/mol/sec)") +
  geom_line(aes(x,y),data=Z,col="blue")
```



```
library(ggplot2)
ggplot(X, aes(X$T..in.K., X$k..in.cm3.per.mol.per.sec.))
+
geom_line() +
geom_point() +
geom_errorbar(aes(ymin=Y1, ymax=Y2)) +
  labs(title="CN+H2 Reaction Rate",x="T (in K)", y="k (in
cm^3/mol/sec)") +
  geom_line(aes(x,y),data=Z,col="blue")
@
\end{document}
```

```
Z          8501 obs. of 2 variables
Values
h          int [1:14] 161 184 174 199 258 274 3…
itemp      num [1:7] 0.00339 0.00258 0.00221 0.…
k          num [1:7] 8535 20444 29379 37717 733…
kfit       num [1:8501] 0.335 0.336 0.337 0.338…
```

```
Degrees of freedom: 7 total; 5 residual
Scale estimate: 0.0411
> exp(12.64633)
[1] 310621.4
> 1060.29508*8.314
[1] 8815.293
>
```

So, and with that we are going to then do the fitting and we are going to do the plotting, so I think it is easier if we have the script and plot the script let us do this, so I am going to get the script from here and that is what we are going to use here. So, one can see that we have we are using rlm() instead of lm() and this here you can say different methods of weighing the errors and then doing the fitting and then this is the fit that you get.

In this case, if you look at the fit parameters, you can see that exponential 12.64633, so it gives you 3.1 into 10 power 5 which is exactly the number that is given in the paper for this fit. And if you look at 1060.29508 multiplied by 8.314, so you get something like 88, the paper still report something around the 9 for this, but this number we are now getting exactly, so I believe it is because of this factor that they have taken into account that they are getting a different number.

And they even give a plus or minus here, how can you calculate such errors and how can you do the analysis is something that I want you to explore to understand what is happening with the data. So, to summarize we looked at in the last session how to look at plotting data, which is linear, in this case now we have looked at data which can be transformed into linear form and then we can do the analysis based on that.

And we have taken two important examples, both are very important in material science and engineering reaction, rates or in the case of silicon manufacturing the processing depends on the rates of diffusion and oxidation et cetera. So, these are the two cases which are amenable to this way of transforming to become linear and once you have linear relationship, of course, you can fit and see how it behaves.

So, we will continue with fitting in the next session, we will look at functions which are not linear, we have already looked at some forms of that, for example we looked at quadratic and we will continue to look at more such nonlinear forms and how to do regression or fitting for these problems. Thank you.