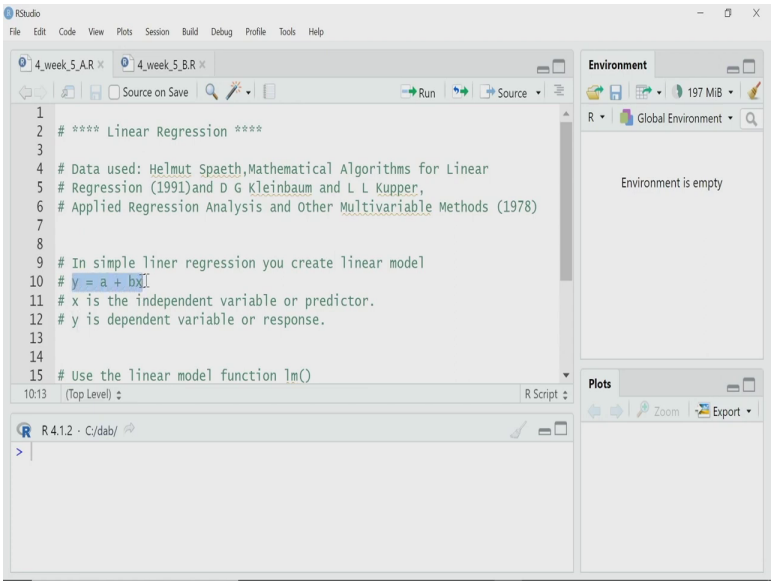
Data Analysis for Biologists
Professor Biplab Bose
Department of Bioscience & Bioengineering
Mehta Family School of Data Science & Artificial Intelligence
Indian Institute of Technology, Guwahati
Lecture 29
Linear Regression Using R

Hello, welcome back. In this lecture, we learn how to perform Linear Regression Using R. In linear regression for example, simple linear regression, we are fitting our data to a linear model. For example, when I mean a linear model in case of simple linear regression, I mean, I have one dependent variable say y or sometime you call it a response variable and it is linearly dependent upon a independent variable x.

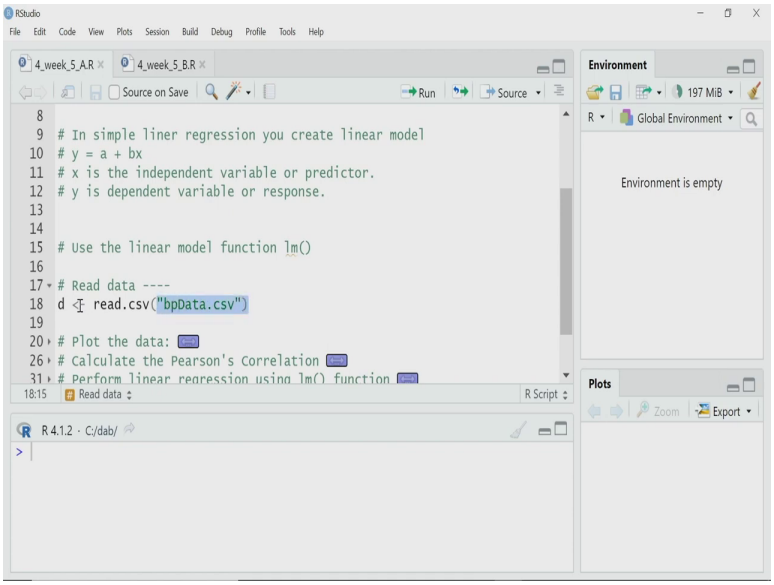
So, your linear model can be y equal to b into x, b is a coefficient or constant plus a. In this lecture, I will start by fitting a blood pressure data where the blood pressure is a response variable and it varies we believe linearly with age of a person. So, I will try to fit this age versus blood pressure data to a linear model of the form y equal to b into x plus a. So, let us do that on R studio.

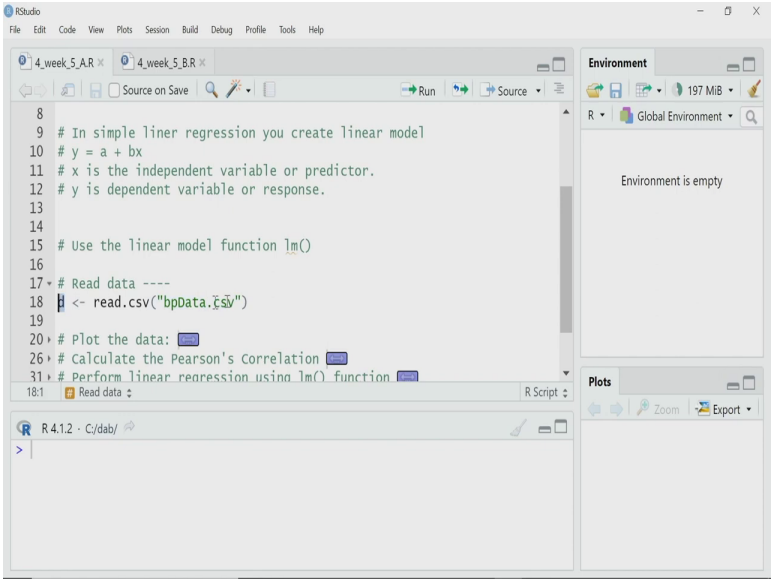
(Refer Slide Time: 01:35)



So, I will be fitting my data to this linear simple linear equation y equal to a plus bx a and b are the coefficients or constant and we have to calculate those values a and b by linear regression. The first step of doing the linear regression obviously, will be to read the data.

(Refer Slide Time: 01:57)

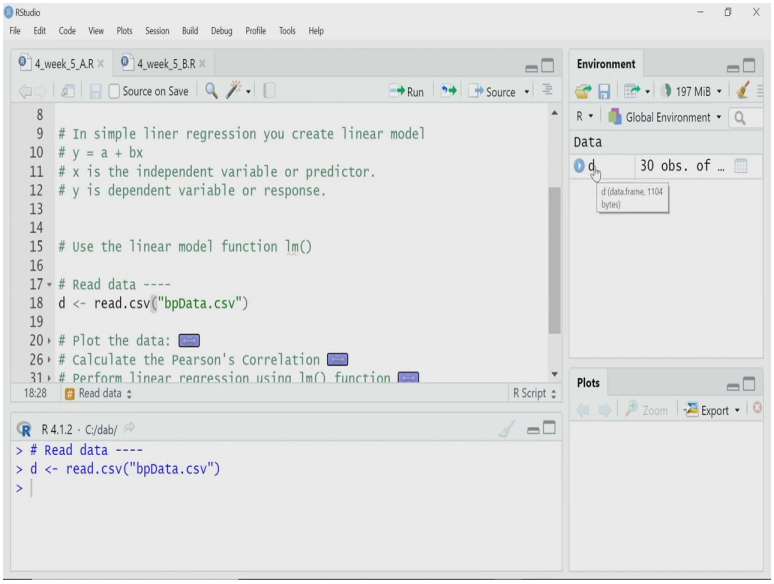


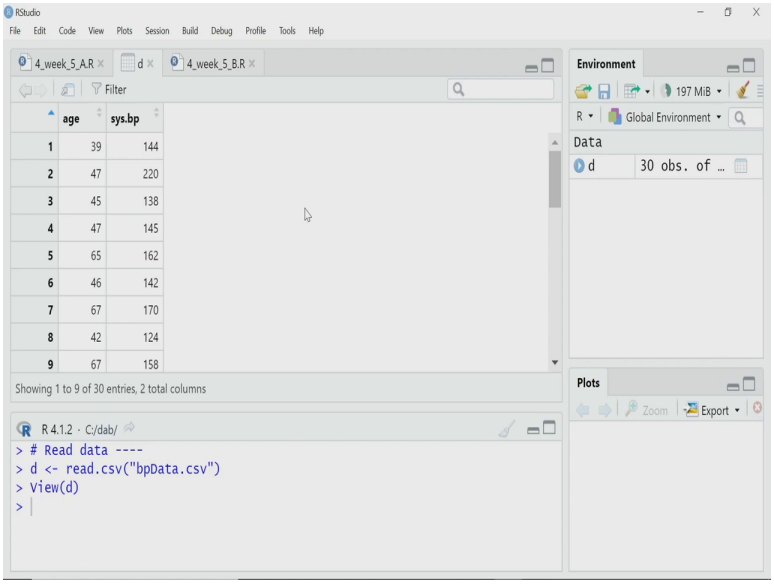


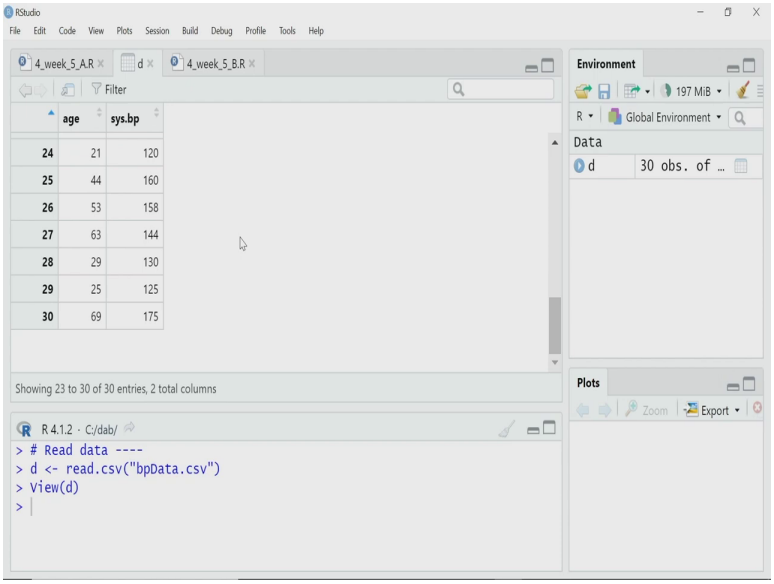
d ← read.csv("bpData.csv")

So, let me read the data using read dot csv file, I have a csv file in my current working directory the name of the file is bpdata dot csv. So, I will read that data using read dot csv function. And I will assign that data to a variable d.

(Refer Slide Time: 02:22)



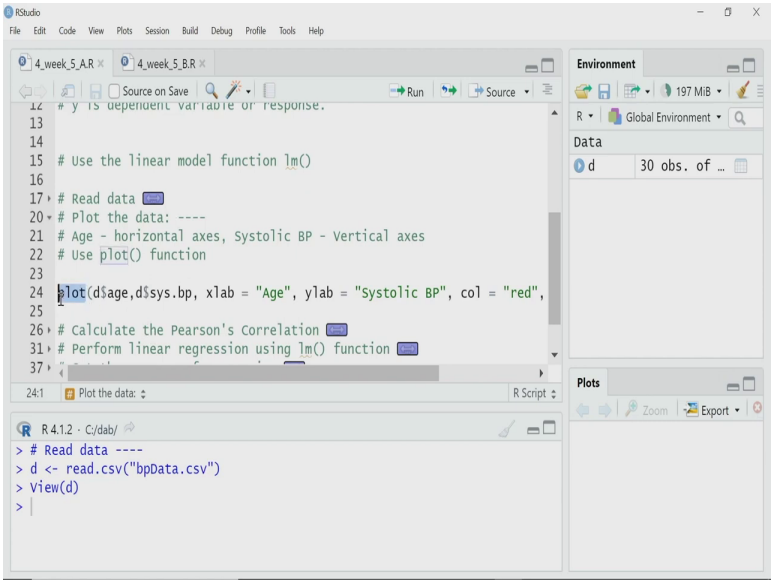


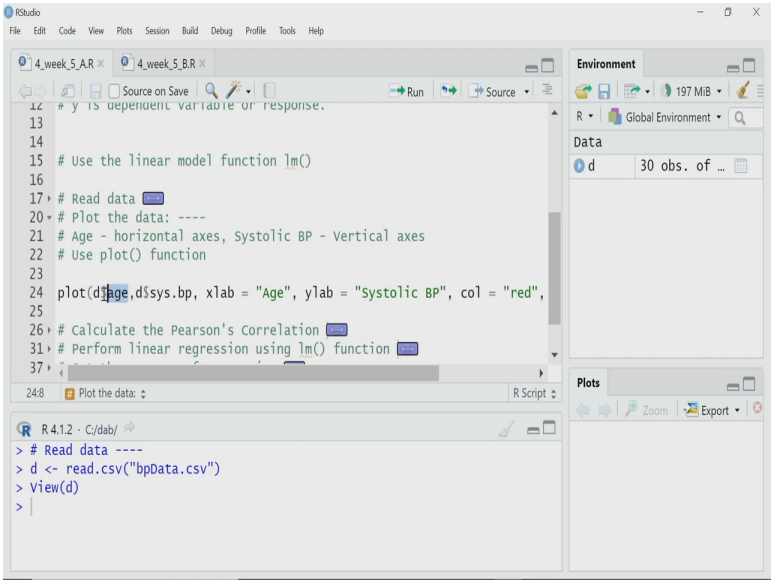


Let us, check the data has been read. And let me click on this data d on environment pane and see what we have in the data. So, it is a two-column data, you can see I have 30 row that means 30 observation. And the first variable the first column is age, age of the person and second column is the systolic blood pressure. So, we believe that systolic blood pressure, which is a response variable has a linear relation with age, so I want to perform the linear regression for that. But before going into performing the linear regression, I want to check whether the data itself has some linear trend or not.

To do that, the simplest way to visually check the data, that means I will plot this data and see whether there is a linear trend between these two variables or not. And the second one that you can do, you can actually calculate the correlation coefficient between this age variable and the systolic blood pressure variable and see whether the correlation value tells us whether there is a linear relation between these two variables or not.

(Refer Slide Time: 03:20)

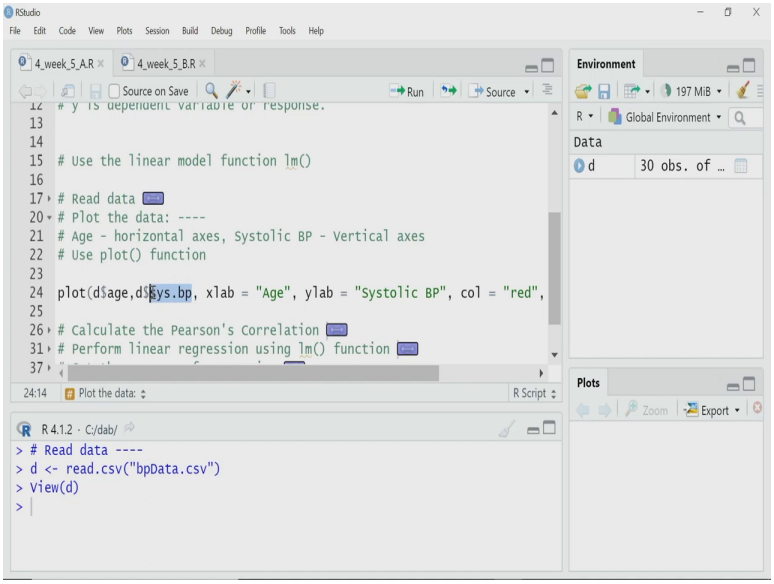


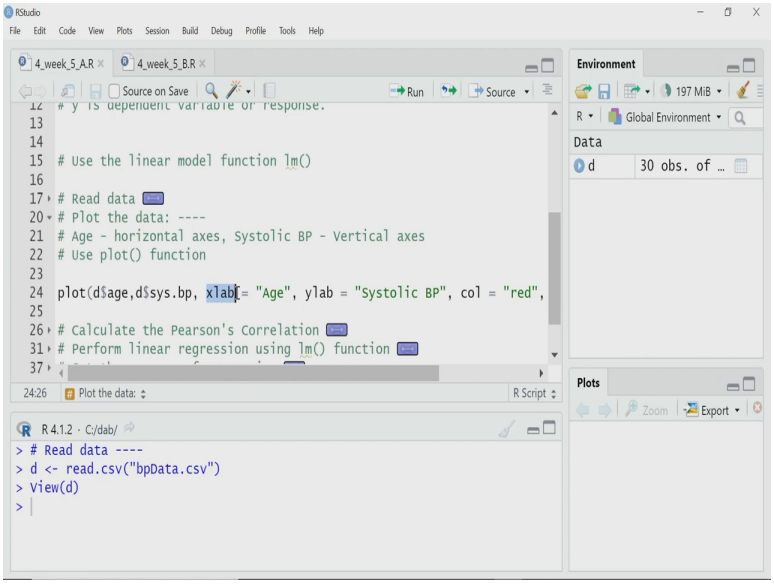


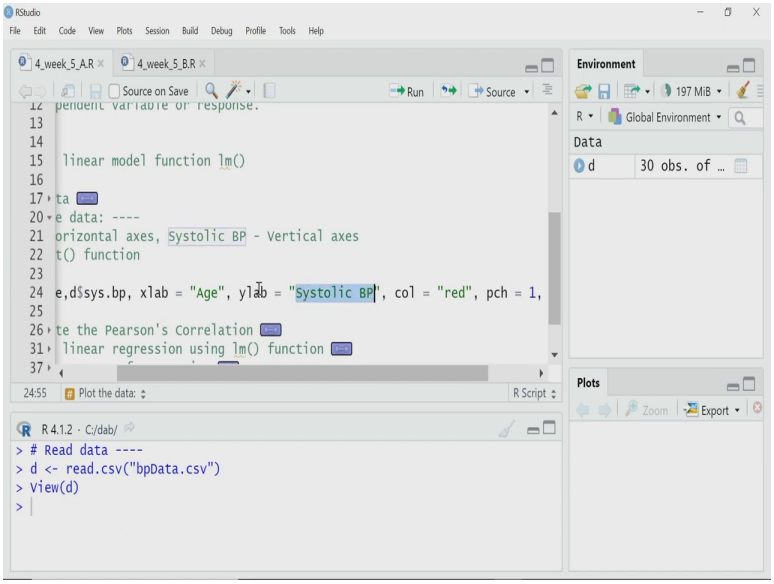
plot(dage, dbp, xlab = “Age”, ylab = “Systolic BP”, col = “red”, pch = 1, cex = 2)

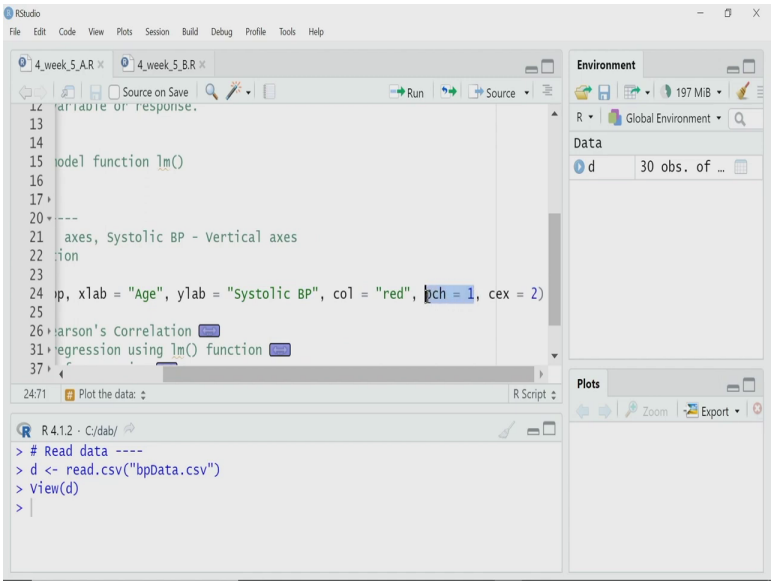
So, I will first draw the data. So, I will plot it as a sort of scatter plot. So, I will use the plot function and it will take multiple input to the multiple argument. For example, age of my, variable age will be my x axis horizontal axis. So, I am writing d, d is the variable where the data is stored, that data frame, d dollar sign, so I am and then putting age I am writing age. So, I want the age variable of d, that is my horizontal axis. That is why it is a first argument.

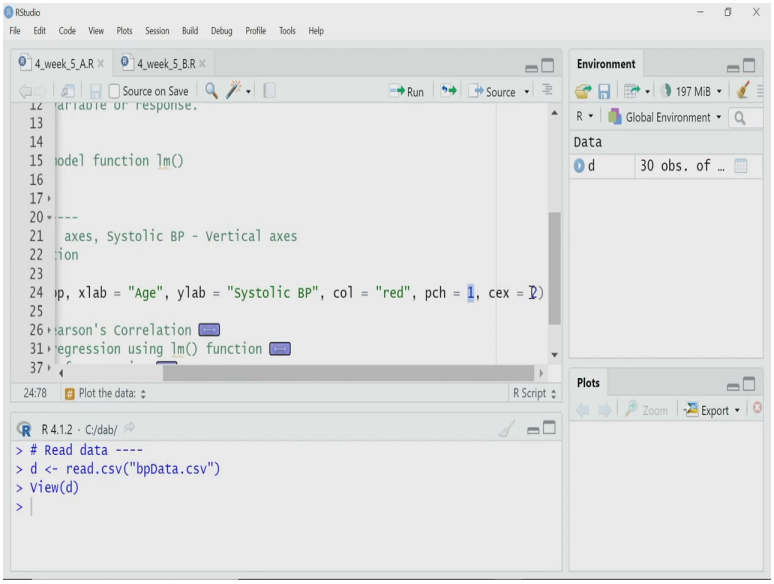
(Refer Slide Time: 03:57)






RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
Source on Save
Run
Source
linear model function lm()
orizontal axes, Systolic BP - Vertical axes
e,d$sys.bp, xlab = "Age", ylab = "Systolic BP", col = "red", pch = 1,
te the Pearson's Correlation
linear regression using lm() function
24:55
Plot the data:
R Script
R 4.1.2 · C:/dab/
> # Read data ----
> d <- read.csv("bpData.csv")
> View(d)
Environment
197 MiB
Global Environment
Data
d
30 obs. of ...
Plots
Zoom
Export

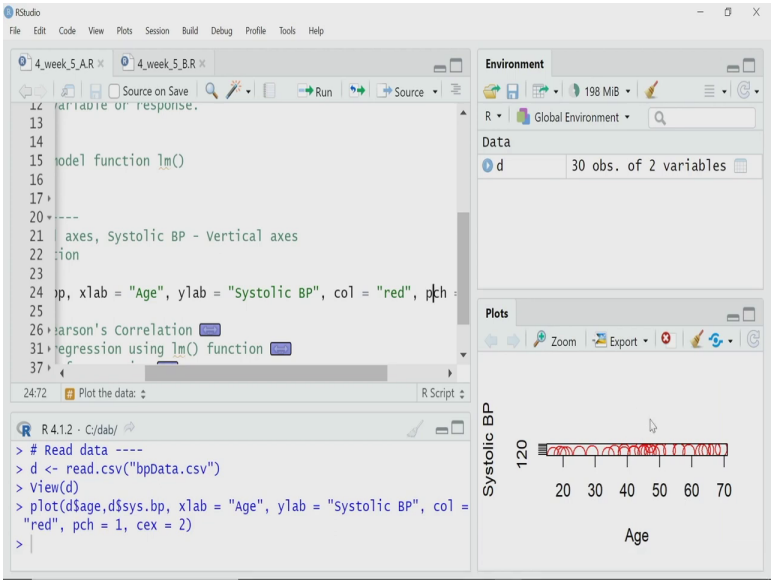

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
Source on Save
Run
Source
odel function lm()
axes, Systolic BP - Vertical axes
p, xlab = "Age", ylab = "Systolic BP", col = "red", pch = 1, cex = 2)
arson's Correlation
egression using lm() function
24:71
Plot the data:
R Script
R 4.1.2 · C:/dab/
> # Read data ----
> d <- read.csv("bpData.csv")
> View(d)
Environment
197 MiB
Global Environment
Data
d
30 obs. of ...
Plots
Zoom
Export



Second is sys dot bp. This is the variable which will go in the vertical axis. So, d dollar sign sys dot bp. This is my vertical axis, then I want to label the x axis as age. So, I am writing x lab equal to Age, I want to label the y axis as systolic blood pressure. So, I have written y lab equal to systolic blood pressure.

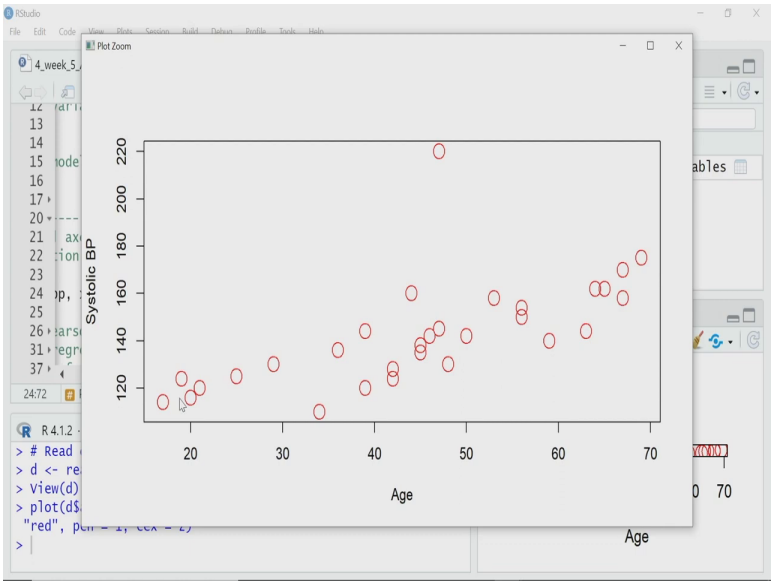
Remember, all these are within the appostrophes, and then I want to use a red colour and for those data points, and I want to use the circle, so the shape of the data symbol that I have to use is denoted by this pch variable and I have assigned that as 1 and cex this argument I have set a 2. So, that will tell us the line thickness of this symbol.

(Refer Slide Time: 04:47)



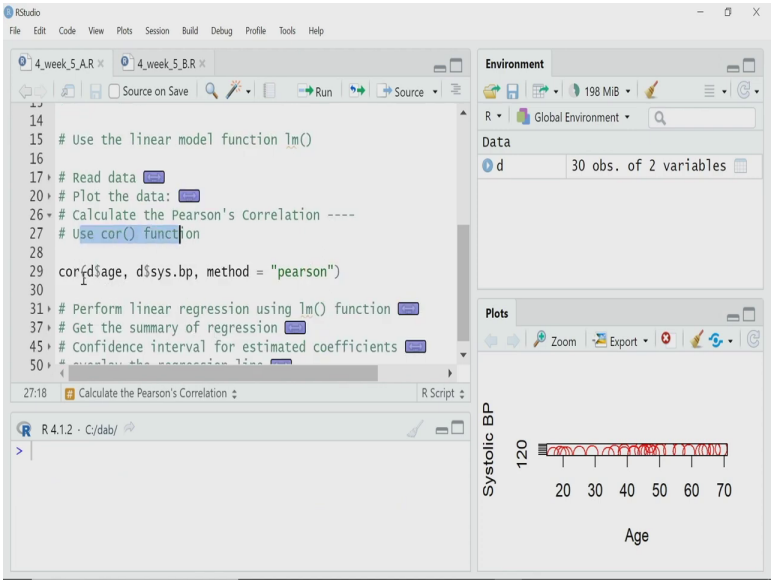
So, now if I plot it, before I plot I will change the size of this plot zone so that it can accommodate it. So, I plot it, you cannot see anything clearly here so I will zoom it.

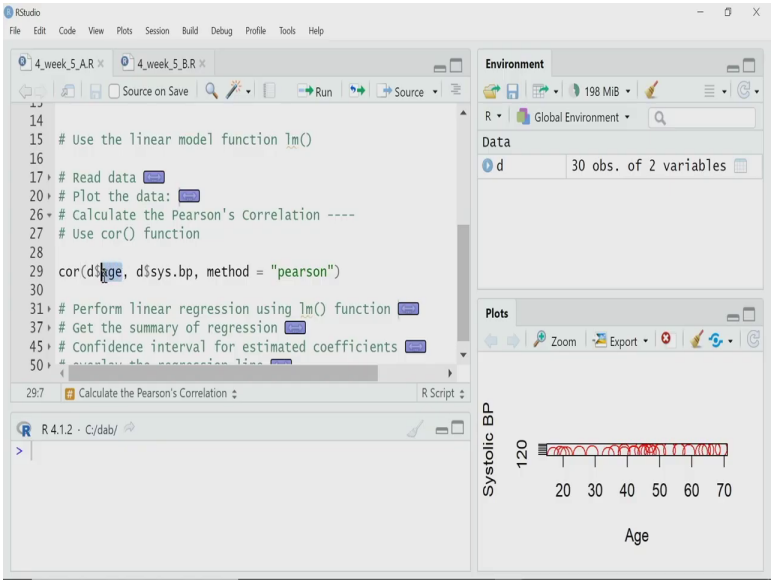
(Refer Slide Time: 05:01)

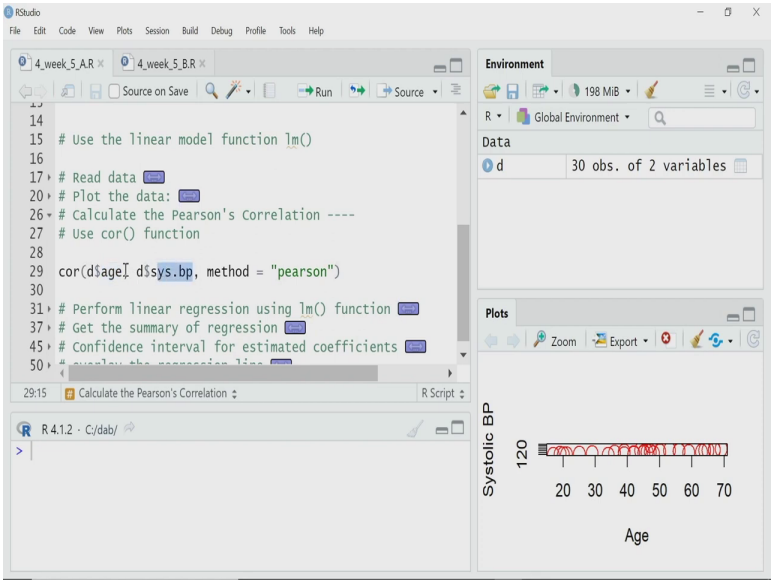


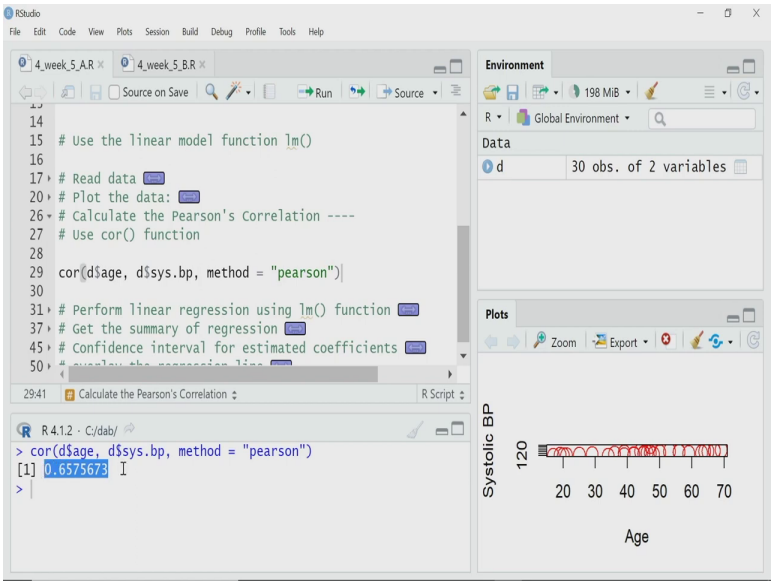
I have zoomed. So, in the horizontal axis, I have age of the person and the vertical axis I have the systolic blood pressure. And these circles are the data points except a outlier here high, whose blood pressure is near to 220 almost all data, you can see there is a sort of linear trend. And it seems as if the blood pressure is slightly increasing as we age as the person age with time that is fine. So, that means the proposition that I will fit a linear model to this data is a right proposition.

(Refer Slide Time: 05:35)









cor(dage, dbp, method = “pearson”)

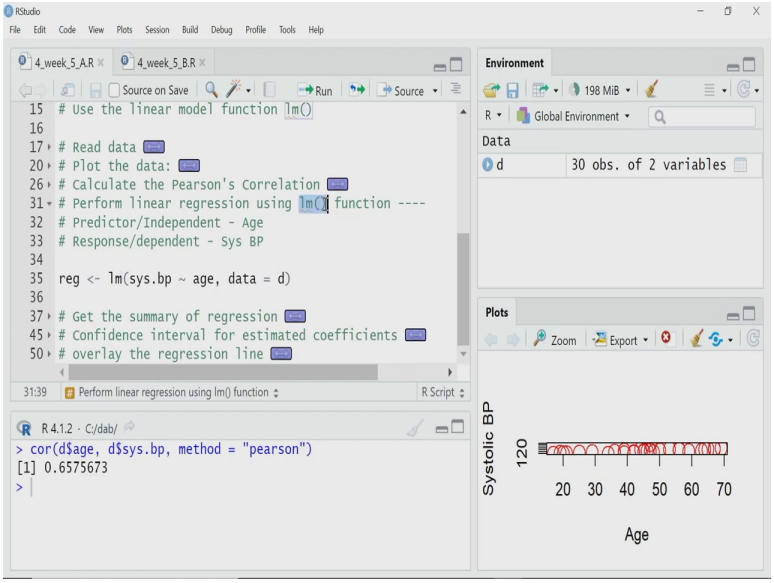
And now I will check the correlation coefficient or rather the Pearson correlation coefficient. So, for that, I will use the function cor I will use the cor function. So, cor function will take multiple argument here, I have to define those two variables, which for which I want to calculate the correlation.

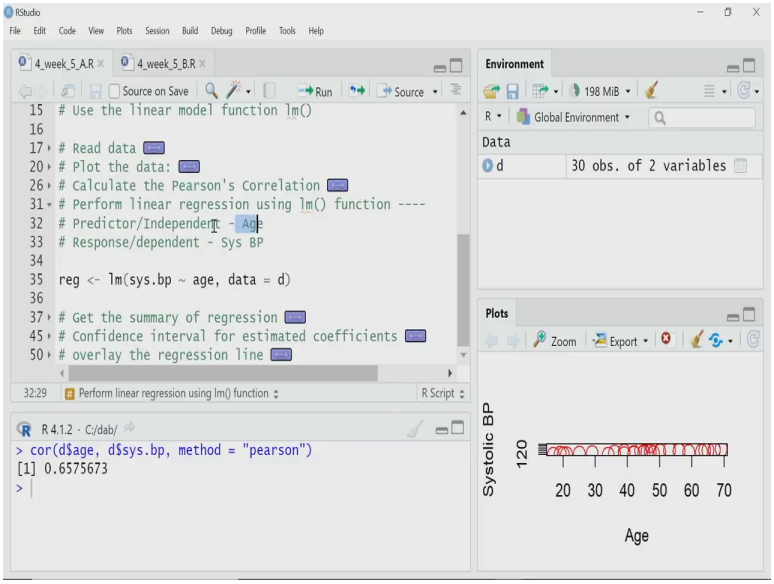
So, I want to calculate the correlation between age of d and sys dot bp of d. So, these two are the variable age and systolic blood pressure. And the method that I want this function to use is

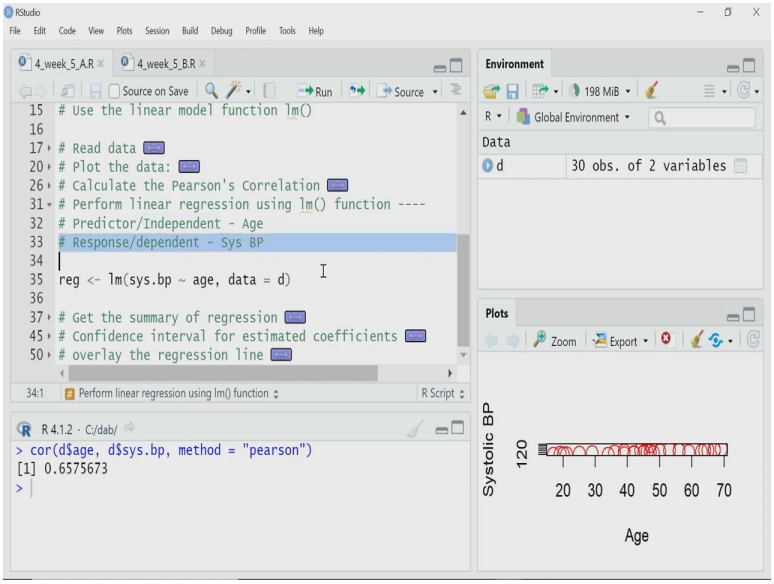
Pearson because it can, this same function can use a correlation between two variables using other methods also. So, I am saying method is equal to Pearson.

So, if I execute this function, it tells me the correlation between age and systolic blood pressure, these two variable in my data, and it is a positive value, and is equal to 0.657. So, that means there is a good positive linear trend between these two variable. So, going we can go forward for the linear regression.

(Refer Slide Time: 06:40)



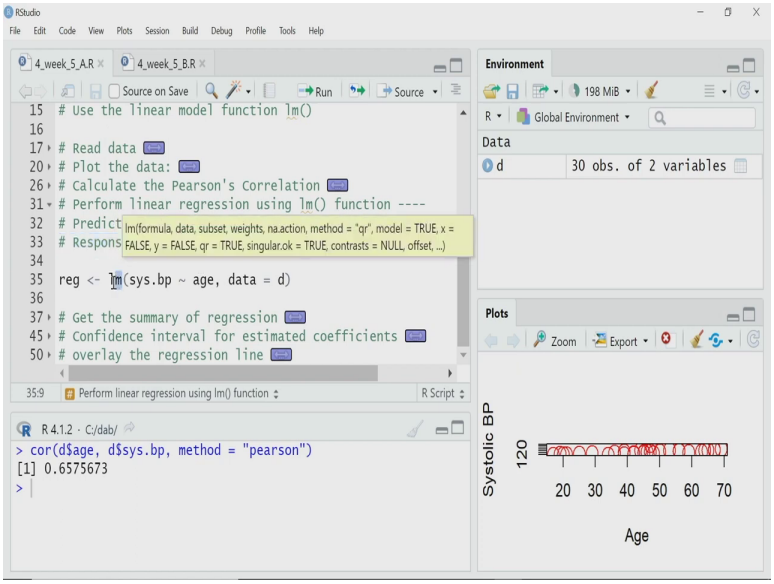


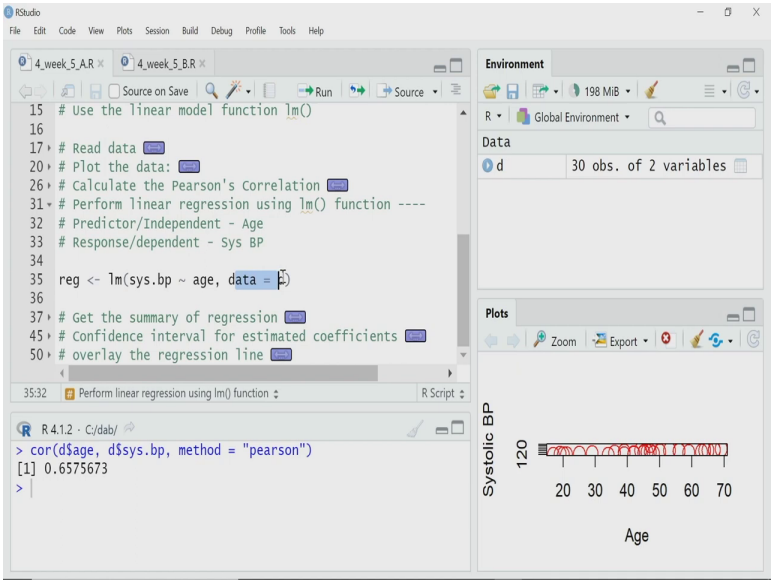


reg ← lm(sys.bp ~ age, data = d)

To perform linear regression, I will use a very powerful function called linear model or lm. So, lm function can do different types of, create different types of linear model and I will go through them one by one in this lecture and another lecture. So, I will use the lm function linear model function and I will use age as my independent variable or predictor, whereas systolic blood pressure will be my response variable or dependent variable.

(Refer Slide Time: 07:13)

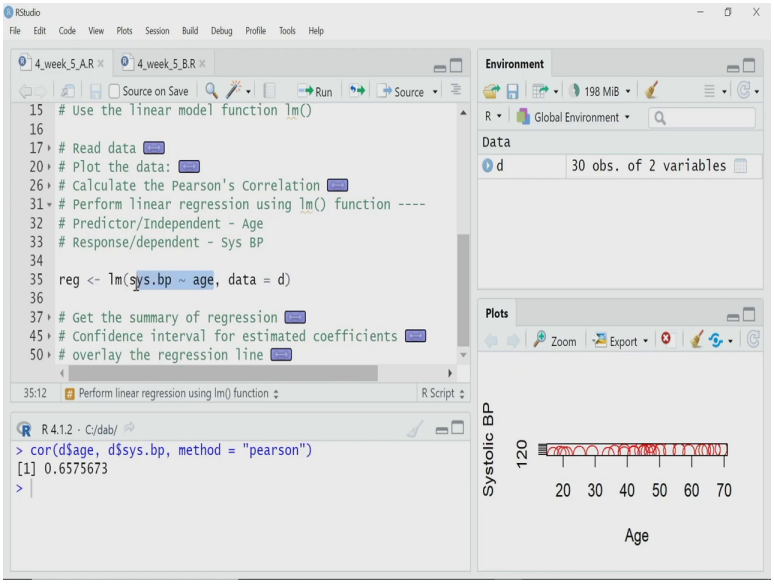


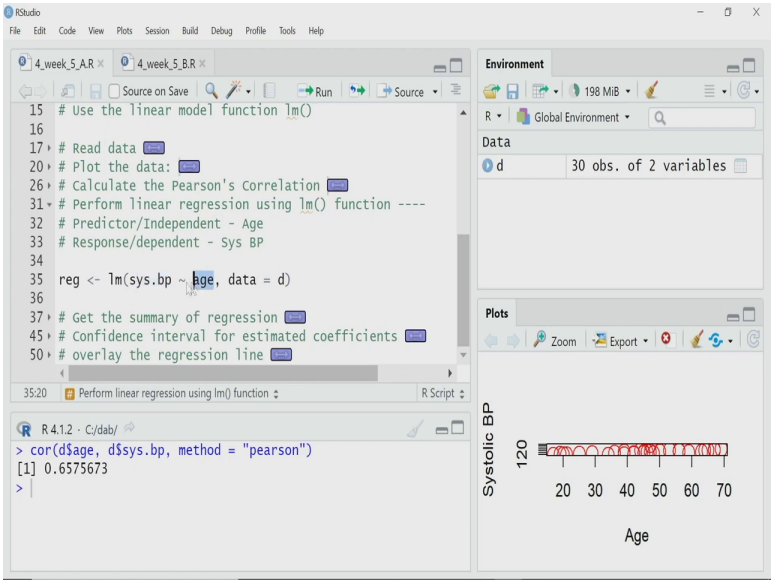


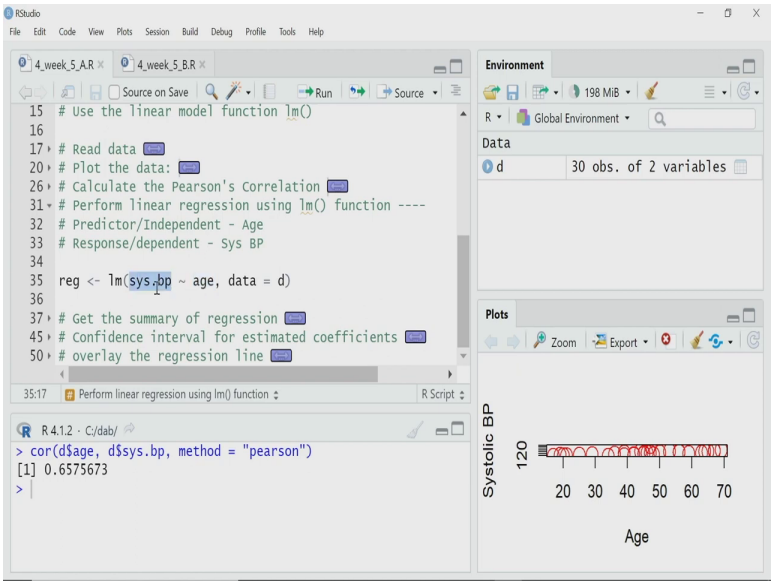
reg ← lm(sys.bp ~ age, data = d)

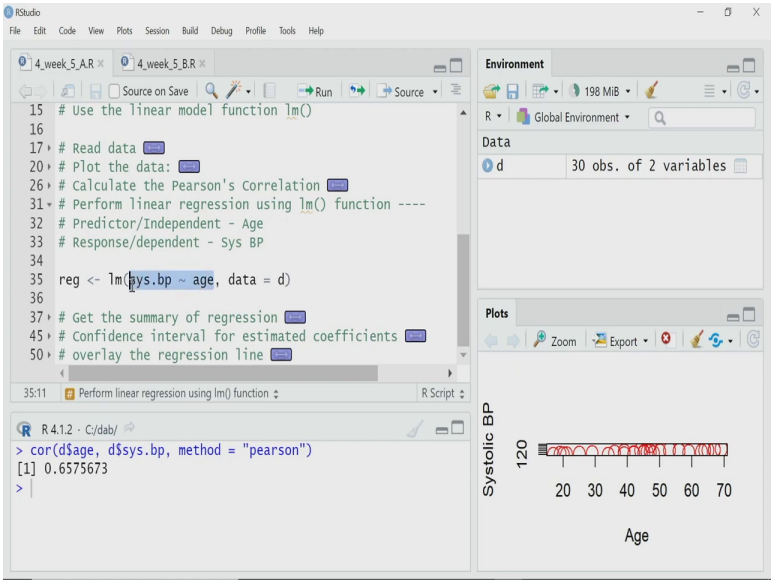
So, how do I define the model? I will call this lm function and I will give two arguments. Obviously, the second argument written here is the data. So, I am saying d is my data, data equal to d. And the first argument defines the linear model that I am creating. So, what type of linear model I am creating, I am creating systolic blood pressure is equal to a constant plus another constant into age y equal to a plus bx.

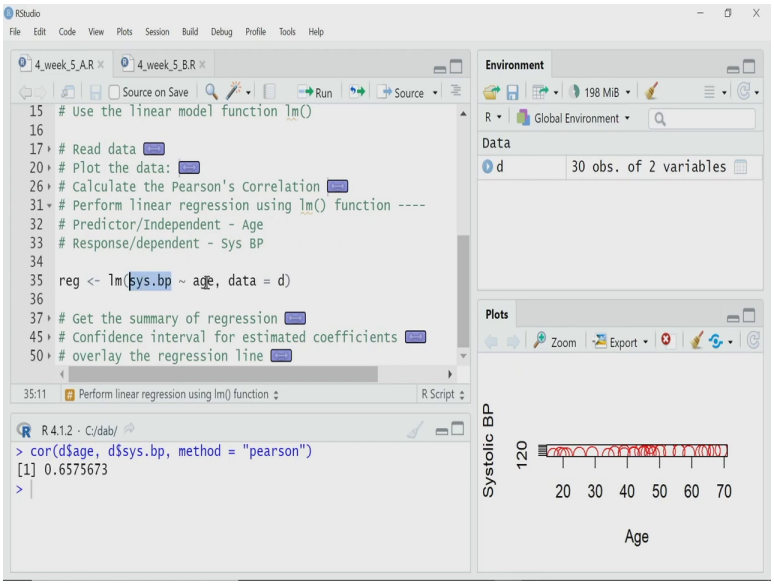
(Refer Slide Time: 07:41)

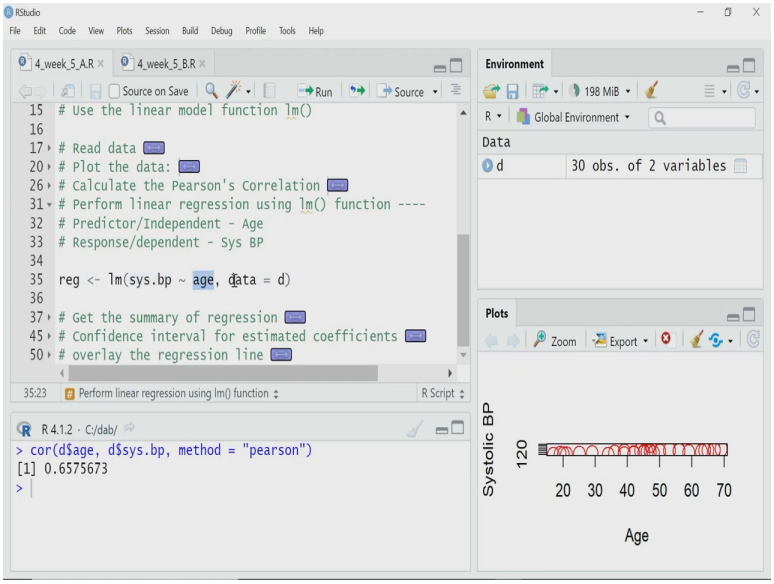



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
Source on Save
Run
Source
15 # Use the linear model function lm()
16
17 # Read data
20 # Plot the data:
26 # Calculate the Pearson's Correlation
31 # Perform linear regression using lm() function ----
32 # Predictor/Independent - Age
33 # Response/dependent - Sys BP
34
35 reg <- lm(sys.bp ~ age, data = d)
36
37 # Get the summary of regression
45 # Confidence interval for estimated coefficients
50 # overlay the regression line
35:20
Perform linear regression using lm() function
R Script
R 4.1.2 · C:/dab/
> cor(dage, dsys.bp, method = "pearson")
[1] 0.6575673
>
Environment
198 MiB
R
Global Environment
Data
d
30 obs. of 2 variables
Plots
Zoom
Export
Systolic BP
120
20 30 40 50 60 70
Age


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
Source on Save
Run
Source
15 # Use the linear model function lm()
16
17 # Read data
20 # Plot the data:
26 # Calculate the Pearson's Correlation
31 # Perform linear regression using lm() function ----
32 # Predictor/Independent - Age
33 # Response/dependent - Sys BP
34
35 reg <- lm(sys.bp ~ age, data = d)
36
37 # Get the summary of regression
45 # Confidence interval for estimated coefficients
50 # overlay the regression line
35:17
Perform linear regression using lm() function
R Script
R 4.1.2 · C:/dab/
> cor(dage, dsys.bp, method = "pearson")
[1] 0.6575673
>
Environment
198 MiB
R
Global Environment
Data
d
30 obs. of 2 variables
Plots
Zoom
Export
Systolic BP
120
20 30 40 50 60 70
Age


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
Source on Save
Run
Source
15 # Use the linear model function lm()
16
17 # Read data
20 # Plot the data:
26 # Calculate the Pearson's Correlation
31 # Perform linear regression using lm() function ----
32 # Predictor/Independent - Age
33 # Response/dependent - Sys BP
34
35 reg <- lm(sys.bp ~ age, data = d)
36
37 # Get the summary of regression
45 # Confidence interval for estimated coefficients
50 # overlay the regression line
35:11 Perform linear regression using lm() function
R Script
R 4.1.2 · C:/dab/
> cor(dage, dsys.bp, method = "pearson")
[1] 0.6575673
>
Environment
198 MiB
R
Global Environment
Data
d
30 obs. of 2 variables
Plots
Zoom
Export
Systolic BP
120
20 30 40 50 60 70
Age

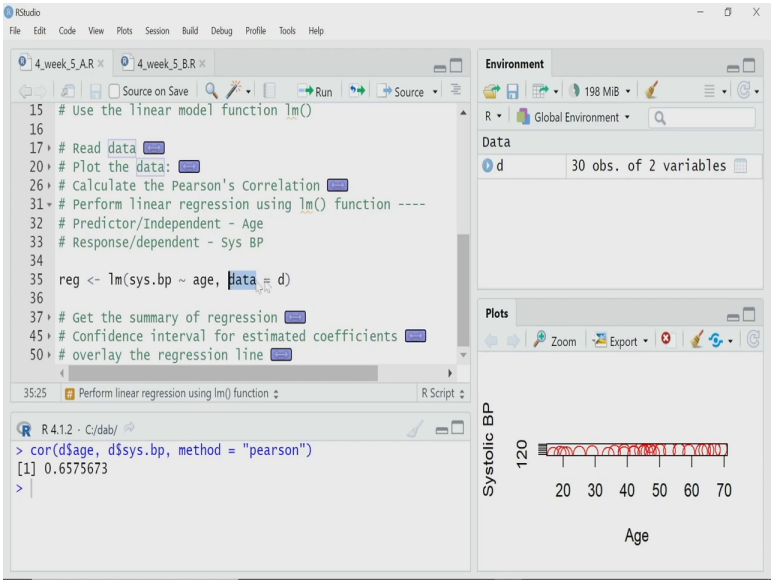

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
Source on Save
Run
Source
15 # Use the linear model function lm()
16
17 # Read data
20 # Plot the data:
26 # Calculate the Pearson's Correlation
31 # Perform linear regression using lm() function ----
32 # Predictor/Independent - Age
33 # Response/dependent - Sys BP
34
35 reg <- lm(sys.bp ~ age, data = d)
36
37 # Get the summary of regression
45 # Confidence interval for estimated coefficients
50 # overlay the regression line
35:11 Perform linear regression using lm() function
R Script
R 4.1.2 · C:/dab/
> cor(dage, dsys.bp, method = "pearson")
[1] 0.6575673
>
Environment
198 MiB
R
Global Environment
Data
d
30 obs. of 2 variables
Plots
Zoom
Export
Systolic BP
120
20 30 40 50 60 70
Age

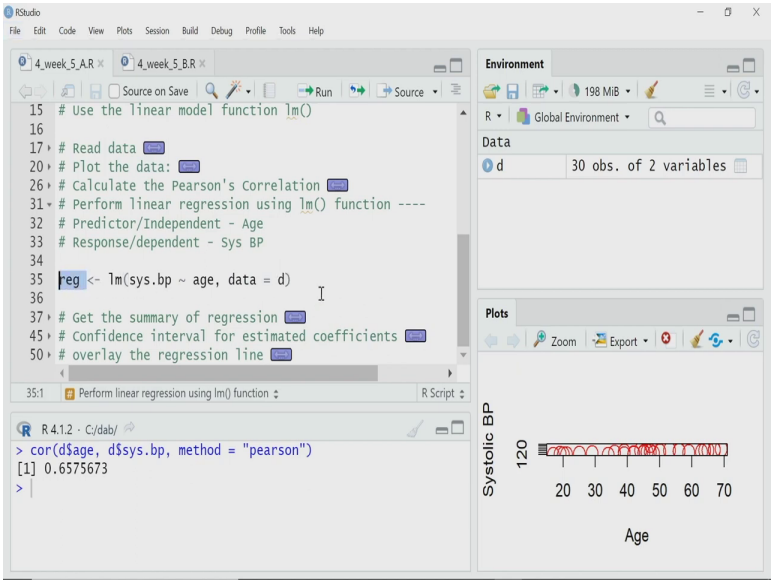


reg ← lm(sys.bp ~ age, data = d)

So, x is age and systolic blood pressure is the y. So, by writing this sys dot bp tilde age, by this way, I am telling the lm function that I want to create a linear model, where the dependent variable is sys dot bp whereas the independent variable is age.

(Refer Slide Time: 08:06)

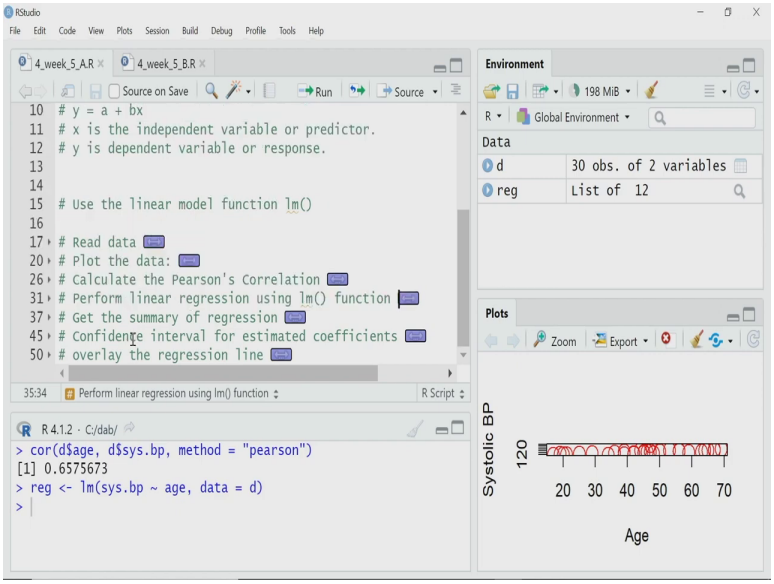


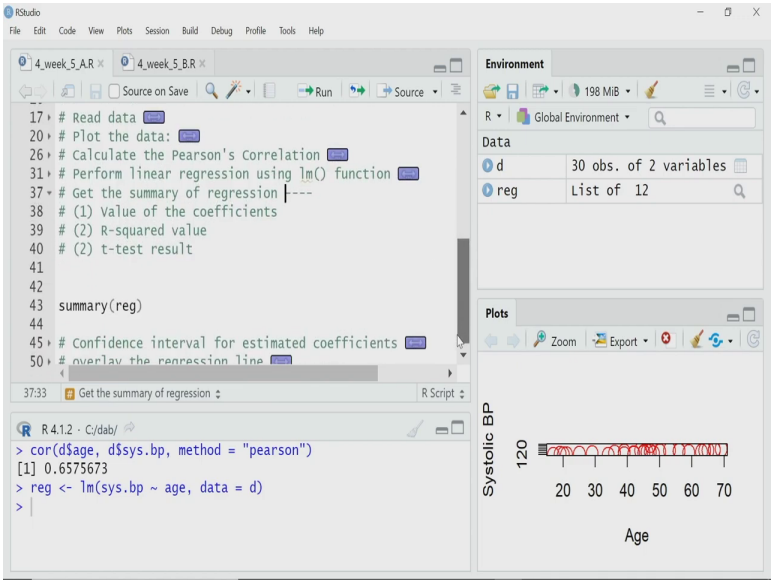


reg ← lm(sys.bp ~ age, data = d)

And the second argument as I said is the data I execute that and take that value, the whatever output comes and assign that output to this reg variable. I have executed. Now, I have to check what I have got by this regression. So, I will say use the summary function to look into what we have stored in the reg variable.

(Refer Slide Time: 08:31)

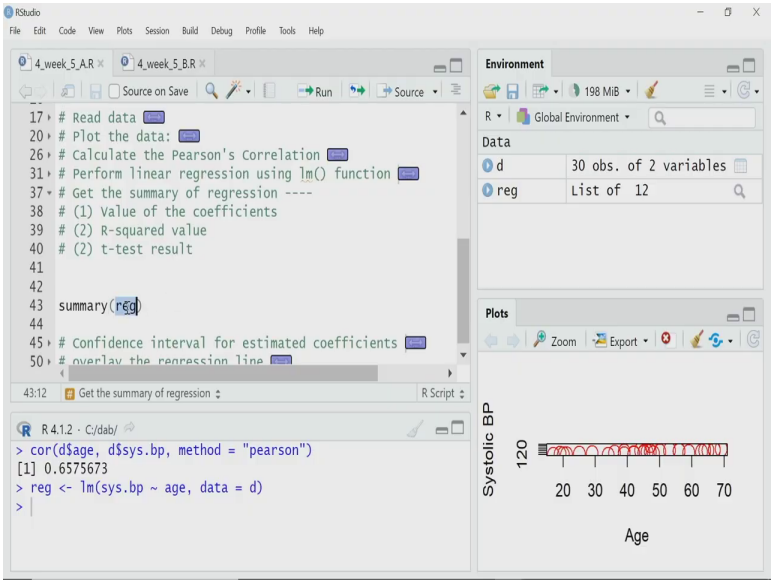




summary(reg)

So, let me check the summary. And this summary tells us lots of things. For example, it will tell me the value of the coefficient y equal to a plus bx, a and b are the coefficients, we want to know them those are unknown to us by regression, we want to find them. So, the summary of this regression data will tell me that what are the value of these two coefficients. At the same time, it will perform t tests for each of these coefficients and tell me whether these coefficients are statistically significant or not. It will also calculate the R squared, which is required to check whether the model is fit properly or not.

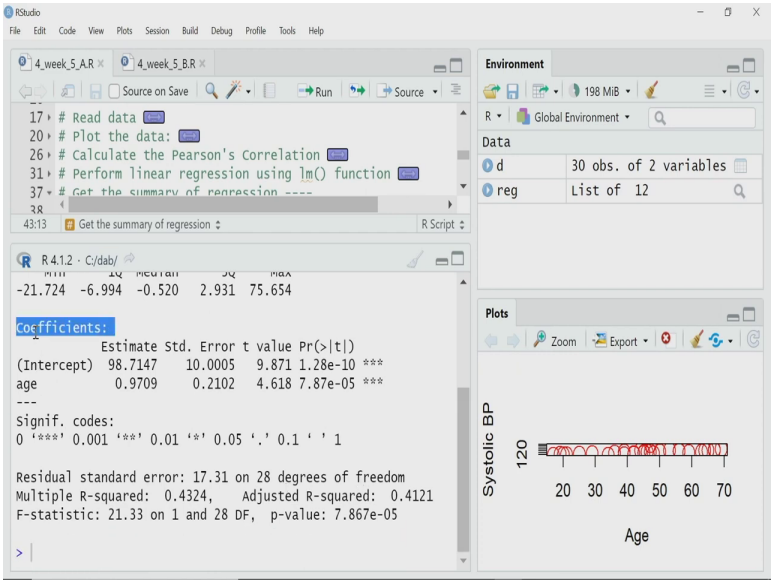
(Refer Slide Time: 09:08)

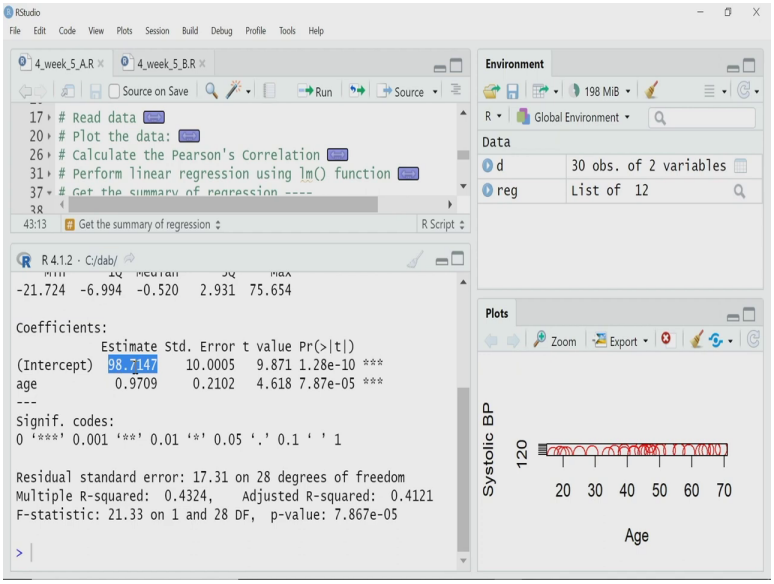


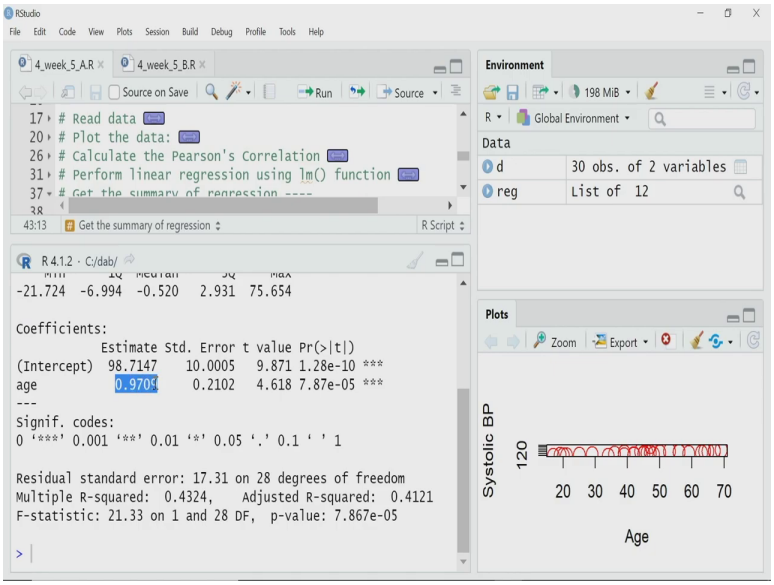
summary(reg)

So, I call the summary function and use reg as the argument because that is where I have stored all the regression data just a few seconds back.

(Refer Slide Time: 09:22)

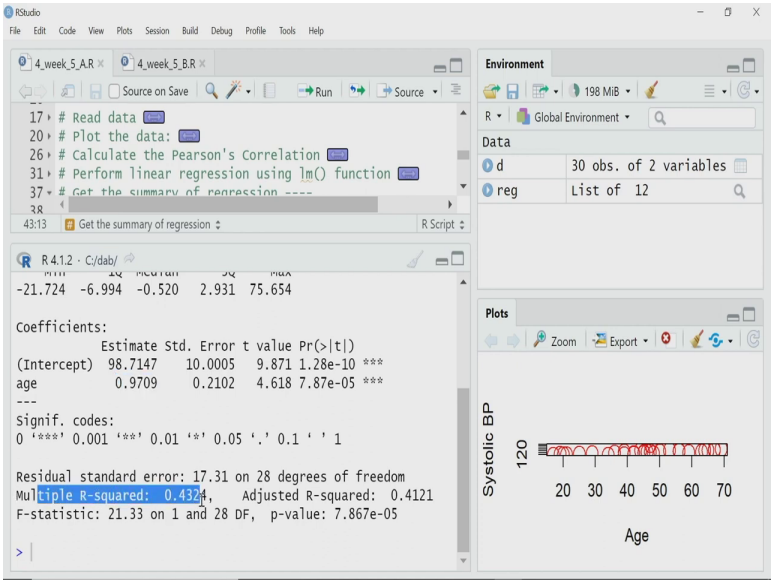


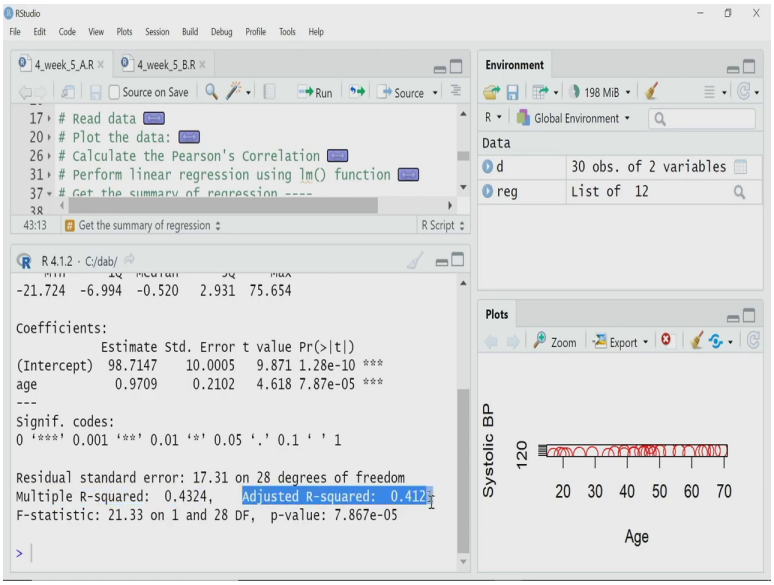




Let me expand. So, here comes all the information given by summary. The most important part I will highlight here, the coefficient, y equal to ax plus b so coefficient are a and b. So, the intercept a is 98.7 and the age coefficient or coefficient for age is 0.97. So, that means my equation in this case is blood pressure equal 0.97 into age plus 98.7. Now, once you have calculated this coefficient, you will obviously go for checking the R squared value, to check the goodness of fit.

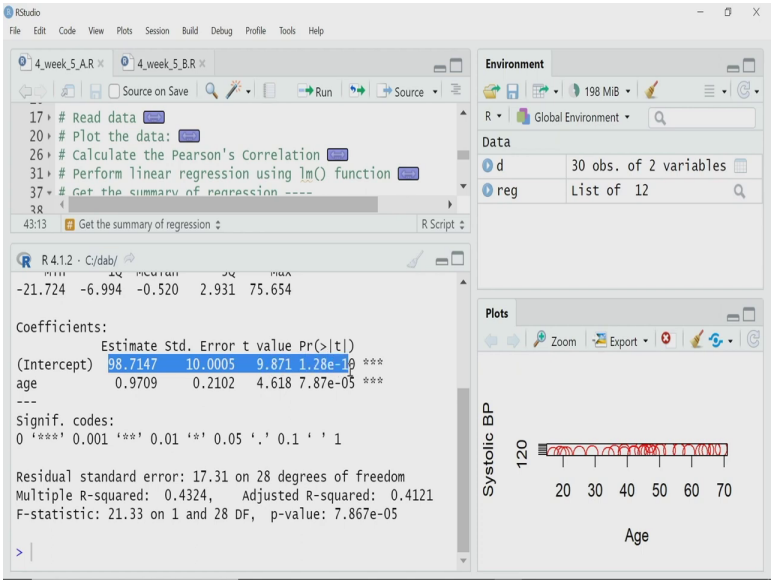
(Refer Slide Time: 10:03)

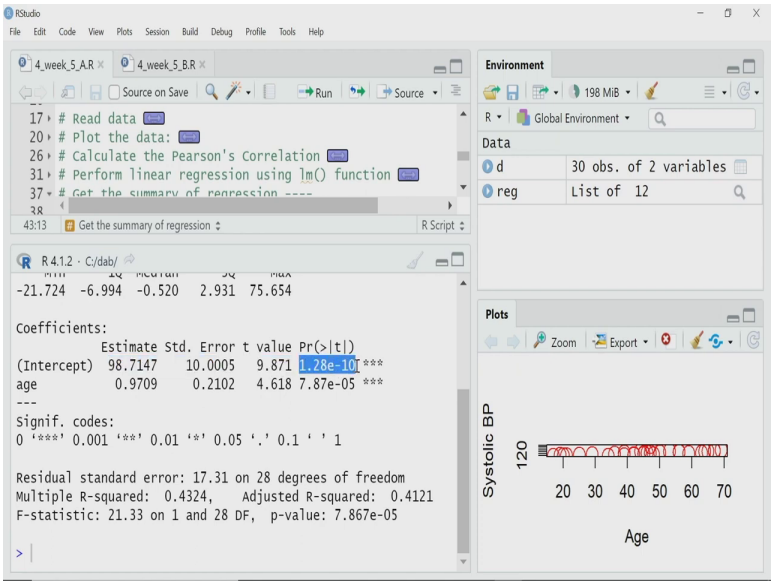




So, the R squared value is given here in this line. And it gives two types of R squared value one is called multiple R squared, and other one is adjusted R square, we have studied adjusted R square where, when we are studying the multiple linear regression adjusted R square is not required for simple linear regression where I have only one independent variable, this multiple R squared is what we call simply R squared value, and I have for this particular problem, I have to check that value. So, R squared value is 0.4324, it is not very close to 1, but looking at the data with the dispersion that we have and considering that it is coming from not a, you know, experiment in lab, but from a population data this quite a reasonable value.

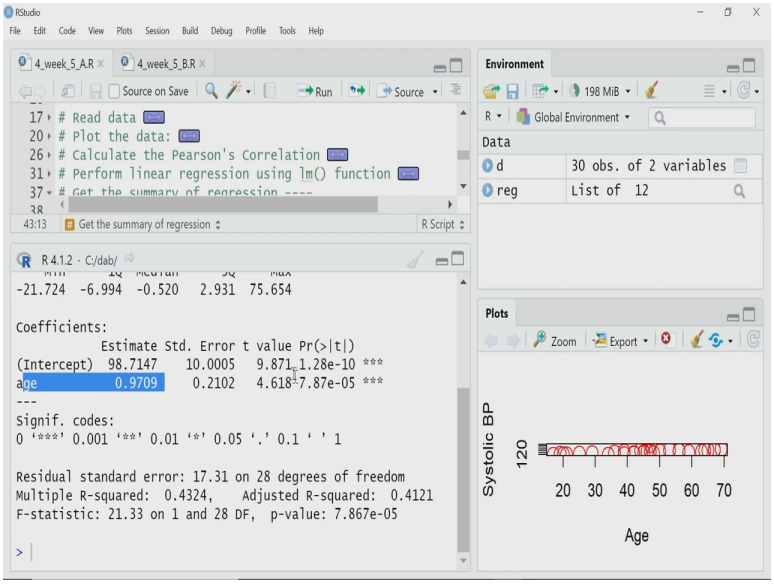
(Refer Slide Time: 11:03)

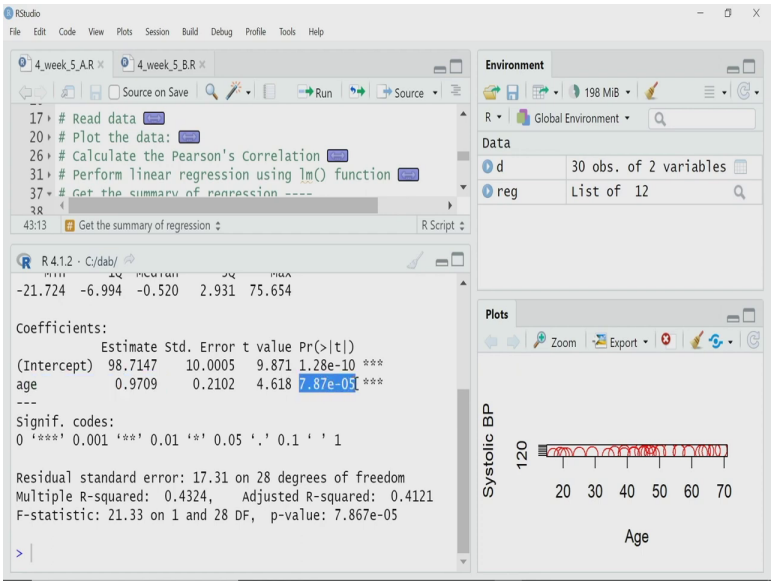




And now I will check for the data for t test. As I said, the lm function has already performed the t test for each of these coefficient, and I will check the probability that is coming for each of them. For the intercept, these first line look at the last column the P is 1.28 into 10 to the power minus 10 extremely small, that means I can easily say that this intercept value estimated by linear regression is statistically significant.

(Refer Slide Time: 11:23)



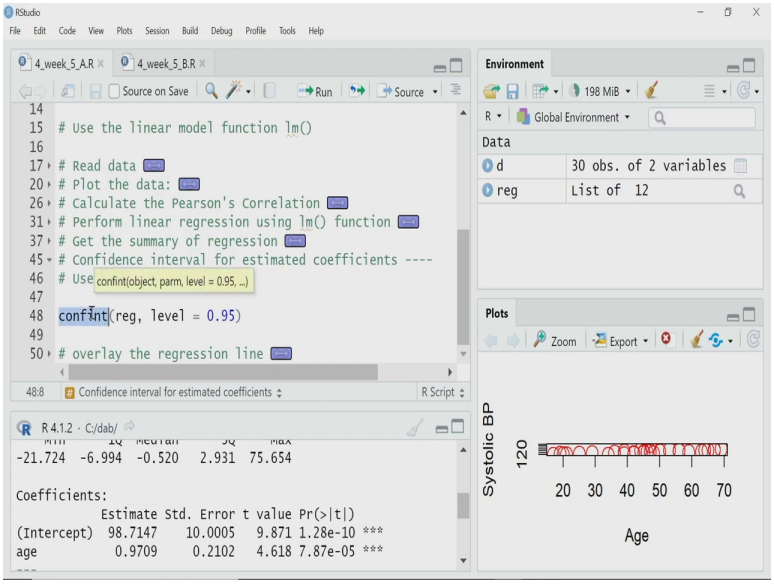


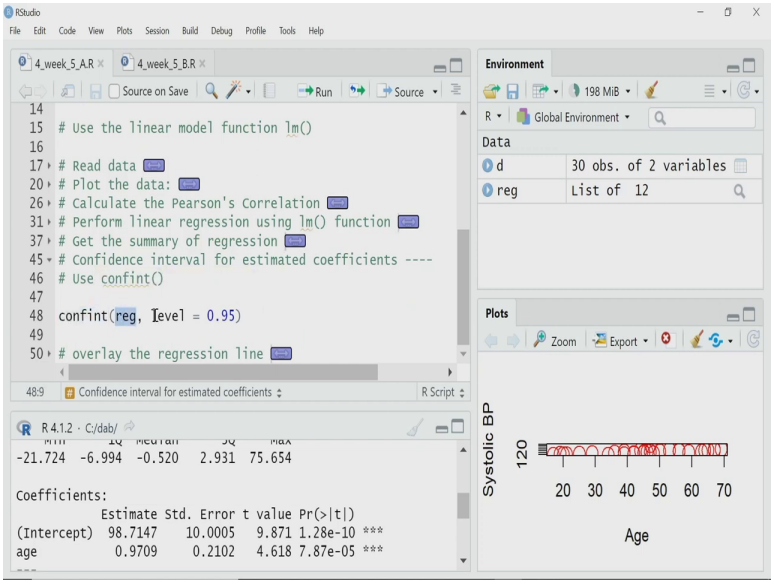
Whereas for the age, the coefficient is 0.97 and the t test result the p value is 7.87 into 10 to the bar minus 5, again, it is very small. That means again, the coefficient, which is just before age, the coefficient for age is also statistically significant, fine. Now, when you use the lm function, it automatically calculates, do the ANOVA.

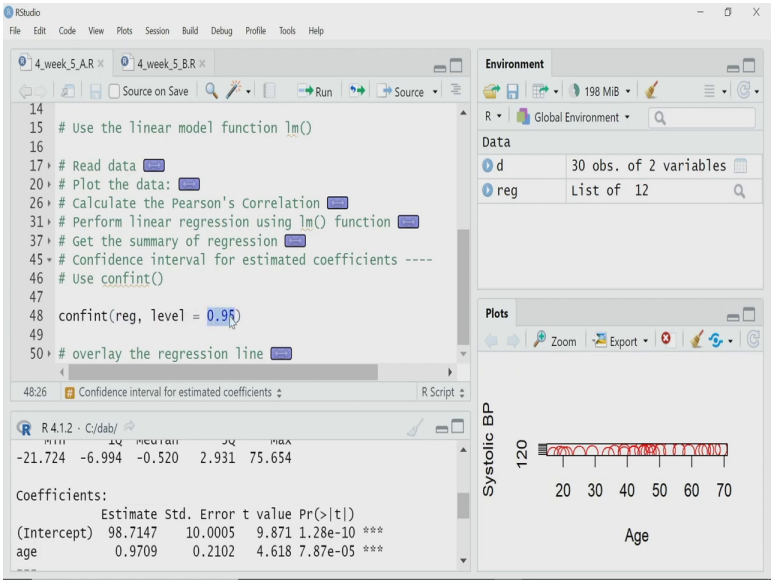
But in case of a linear regression, simple linear regression with one independent variable ANOVA does not make sense here, but that will be useful when I will use the same lm function for the multiple linear regression. So, that is what I have got from the summary of this regression result. And we are almost done with that a few things that I have to check again, you may be

interested to know the confidence interval for this estimated coefficient. So, I will calculate the confidence interval.

(Refer Slide Time: 12:24)



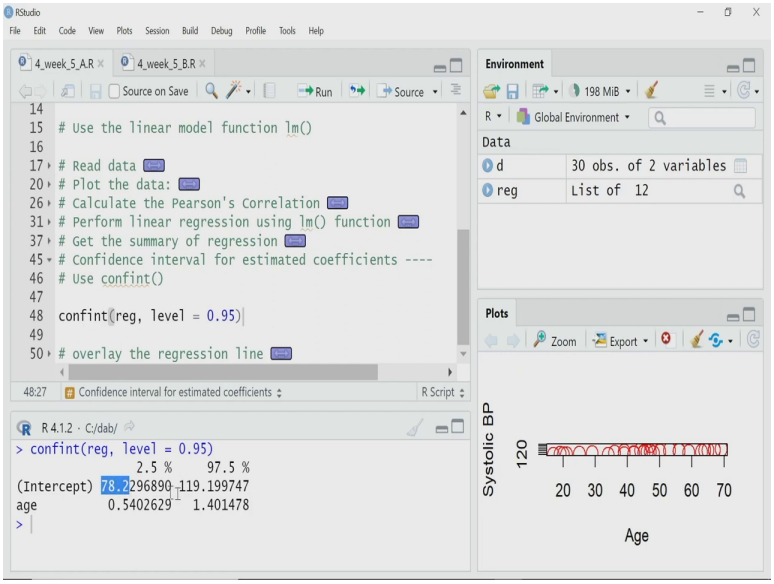


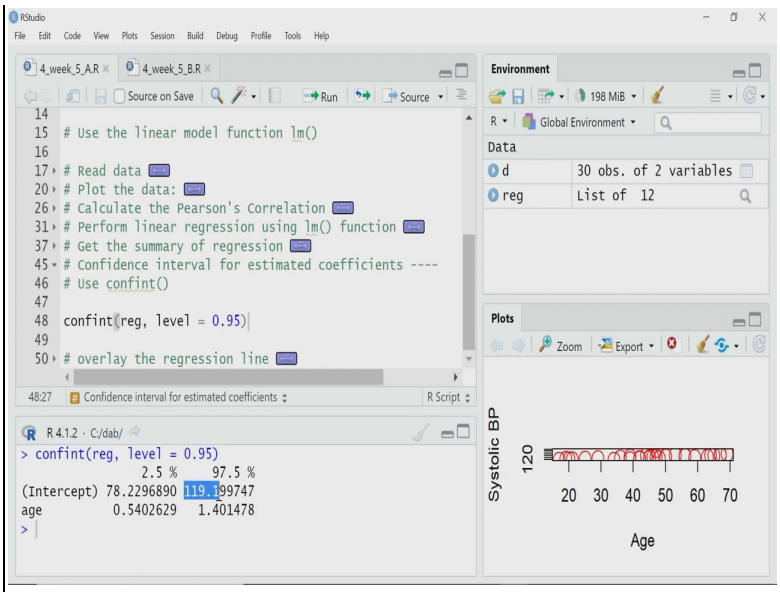


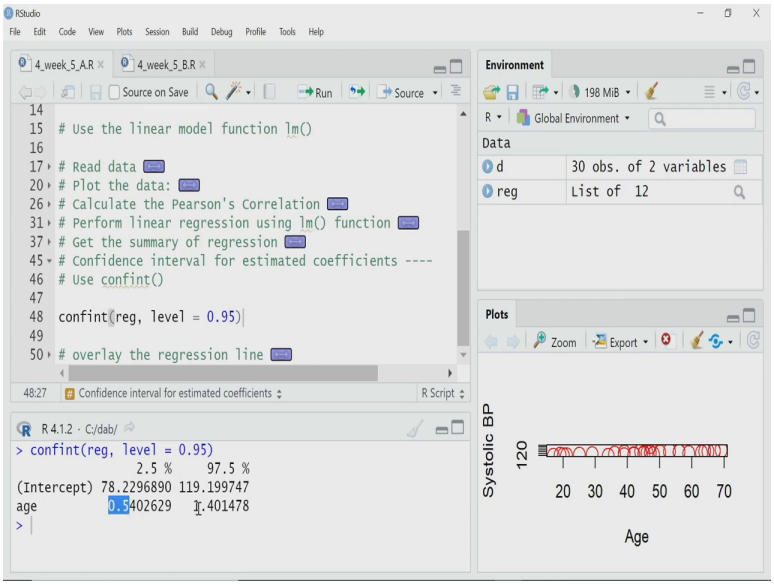
confint(reg, level=0.95)

To calculate the confidence interval, I will use a function called confint. And I will call that function confint and I will give the reg variable which is storing all the regression data as one argument. And I will define the level of significance and I am setting 95 percent that means 0.95. So, let me clean it further the console a bit. And if I call this confint function, it will calculate the confidence interval for each of the calculated coefficient.

(Refer Slide Time: 13:00)



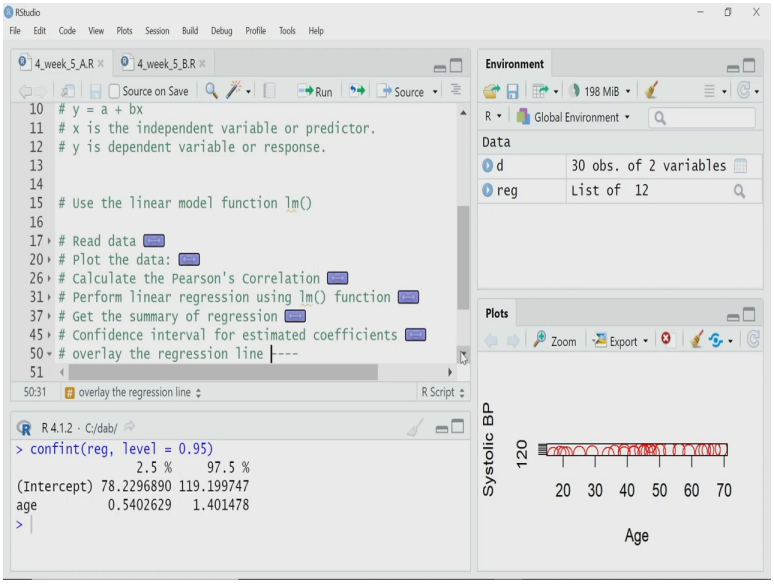


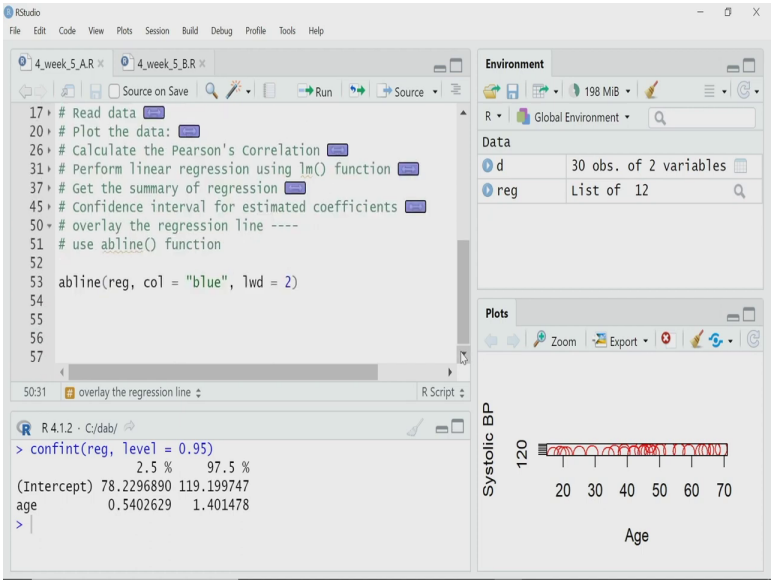


confint(reg, level=0.95)

So, it is saying that at the confidence level of 95 percent that is 0.95 level the value true value of the intercept varies from 78.2 to 119.19 whereas for the same confidence interval, the age the true value of the coefficient for age varies from 0.5 to 1.4. So, easily using this confidence interval function confint I can calculate the confidence interval of my estimated parameters or estimated coefficient. Now, I am done with the regression, I have got the coefficient I have done that calculate the confidence interval of those I have checked that they are statistically significant. Now, the last step remains with me is to actually plot this regression line and overlay on the original data set.

(Refer Slide Time: 13:55)

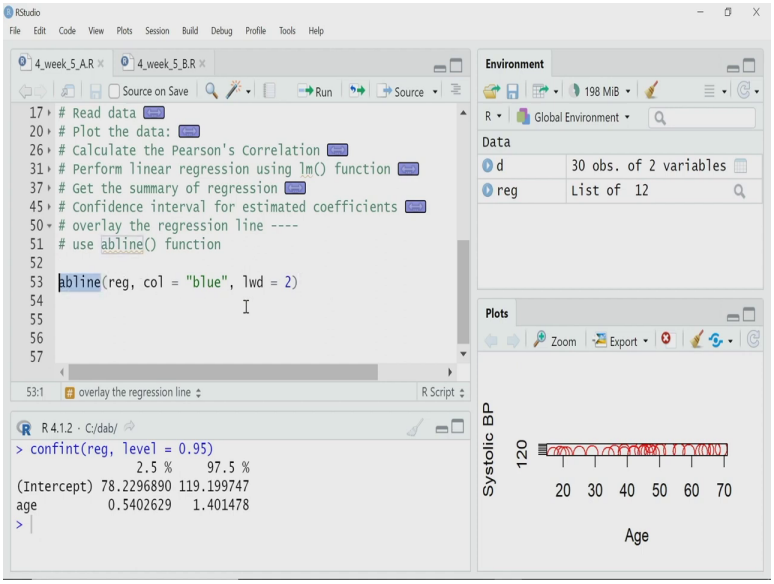


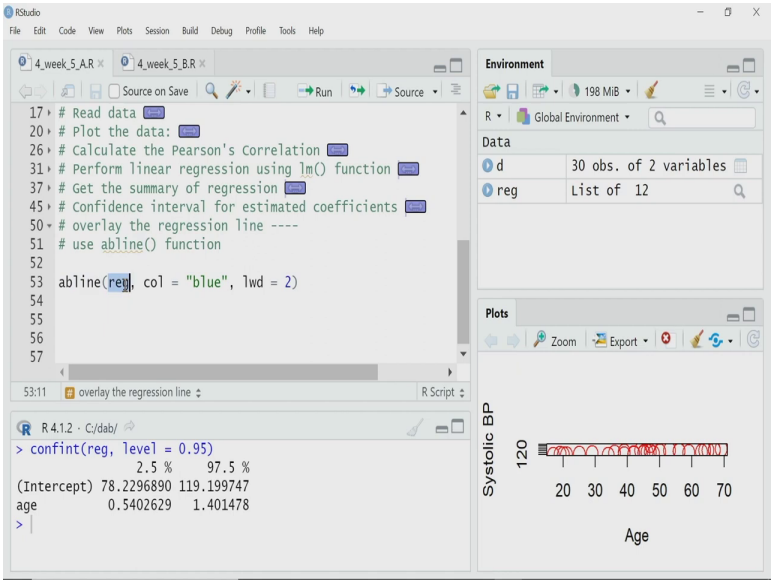


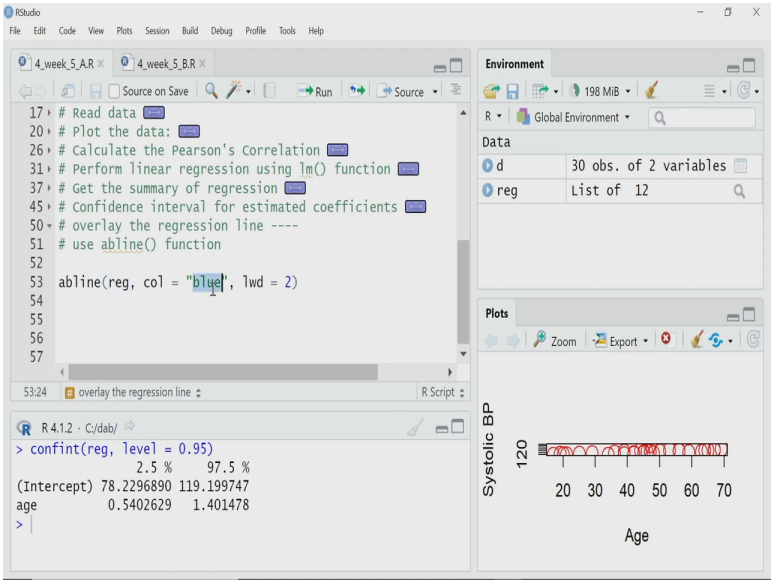
abline(reg, col= ”blue”, lwd = 2)

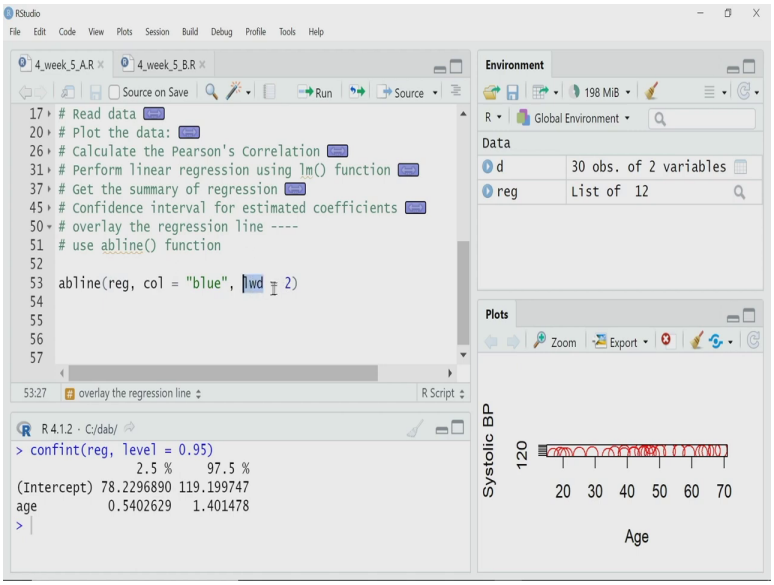
So, how I will do that? I will now overlay my regression data, the regression line that we can generate using the calculated coefficient and the coefficient for age and intercept those values.

(Refer Slide Time: 14:12)





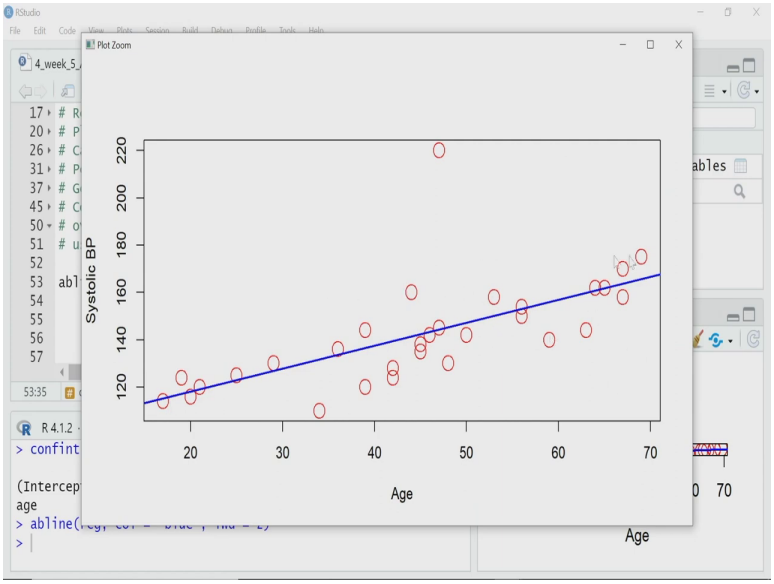




abline(reg, col= ”blue”, lwd = 2)

So, what I will do, I will use the abline function, abline function add line on the existing plot. So, abline function, I will give this reg variable which is storing all the regression data as one of the argument. I want a blue line so I am saying colour equal to blue, and the line width, or the thickness if you say is lwd, I have defined a 2. So, if I execute that, the line has been drawn.

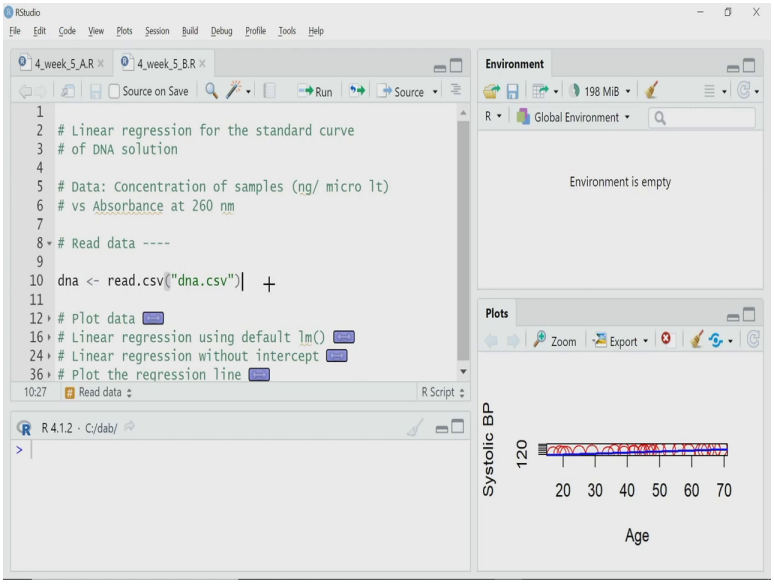
(Refer Slide Time: 14:39)



But let me zoom to see it clearly. So, now you can see this red are the scatter for original data. And this regress line is this blue line. That is all so simple. So, I have used the lm function to do perform the regression, and then got the statistics and as I am happy, I am plotting the data. Now, I will move into another example for linear regression, and that is one example that most of you working as a biology student or must be performing that type of regression regularly. That is for estimating the concentration of a unknown solution of DNA or protein.

So, if you do some calorimetric or UV visible spectroscopy to measure the concentration of either DNA or protein, what you do, you create a standard curve, you have a experiment. For example, suppose you have known samples of DNA with different known concentration, you take a UV visible spectrophotometer and measure the absorbance of those samples, and you plot that data perform a linear regression to get a standard curve. And then you use that standard curve to estimate the concentration of unknown DNA sample using its absorbance. So, you must be doing that. So, I will show how you can perform the linear regression for that type of experiment using R.

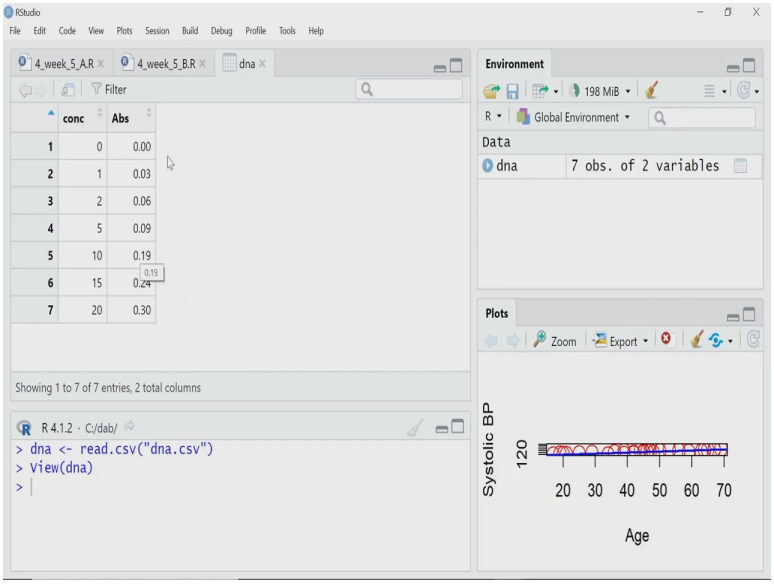
(Refer Slide Time: 16:01)



dna ← read.csv("dna.csv")

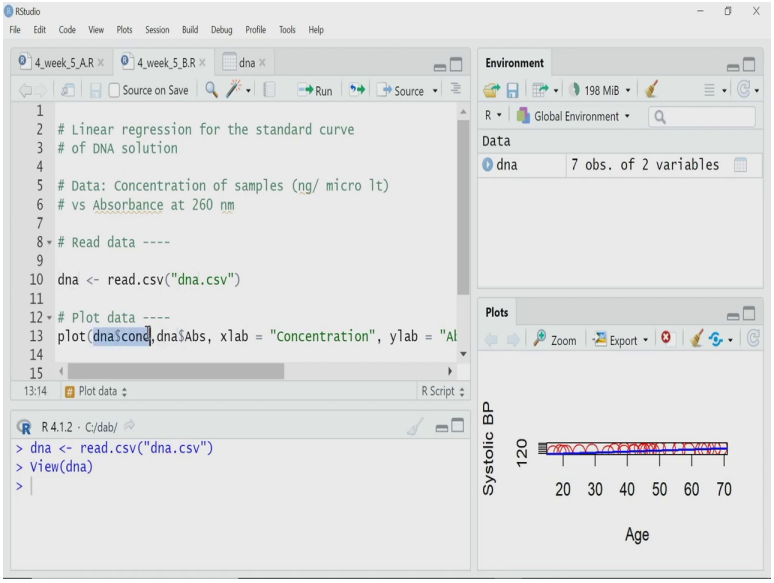
So, I already have a data set where I have measured taking different samples of known concentration of DNA and then I have measured the absorbance of that at 260 nanometre. So, let me first read that data, and this is a csv file called DNA dot csv present in my working directory. So, I will use the read dot csv function to read that and assign that data to a variable called DNA. Let me show you what is the data I have.

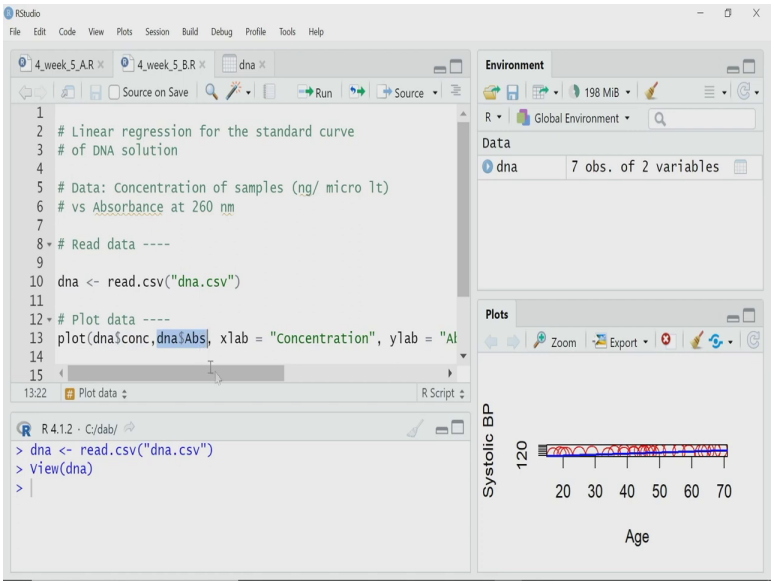
(Refer Slide Time: 16:30)

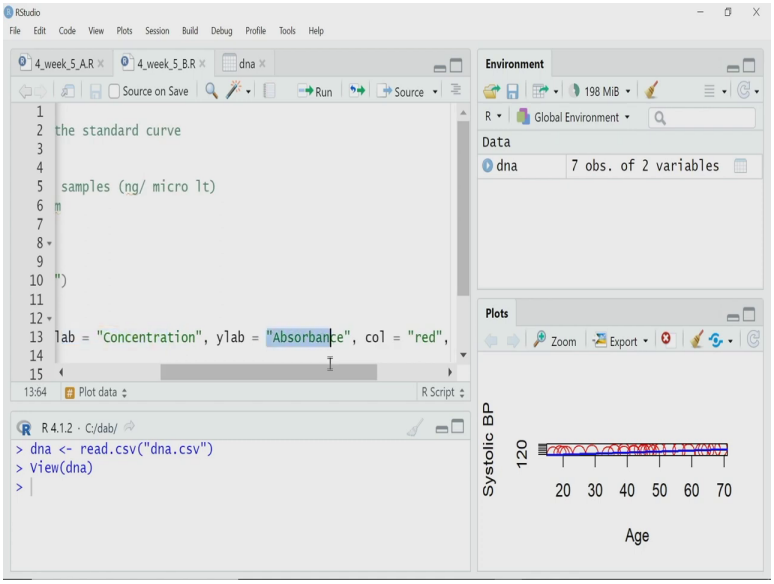


So, this is my data concentration versus absorbance. So, I have the concentration in the first column and the absorbance in the second column and I have 7 samples and I will highlight one point the first sample is concentration 0 and absorbance is 0. That is what our usual practice, we take a blank and then we auto 0 the reading of my machine. So, by default by this auto 0, you are making the assured that the absorbance becomes 0 when the concentration of the DNA in the sample is 0.

(Refer Slide Time: 17:05)



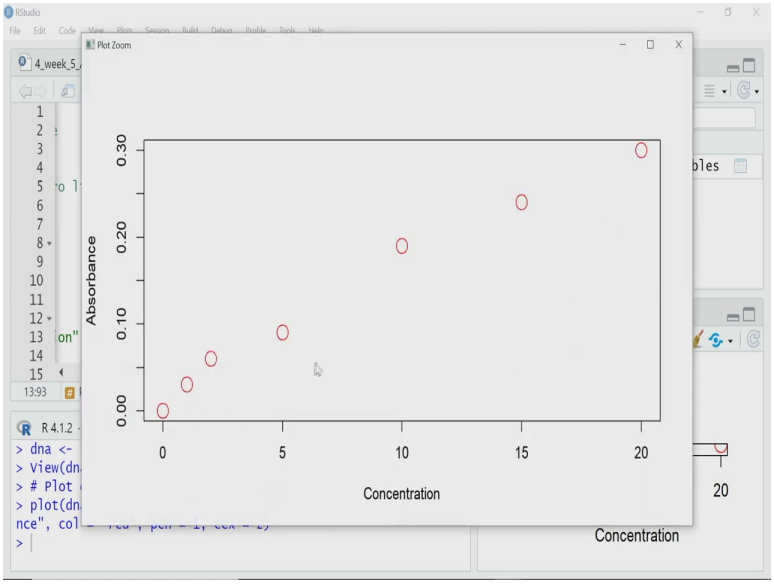




plot(dna$conc, dna$Abs, xlab = “Concentration”, ylab = “Absorbance”, col = “red”, pch = 1, cex = 2)

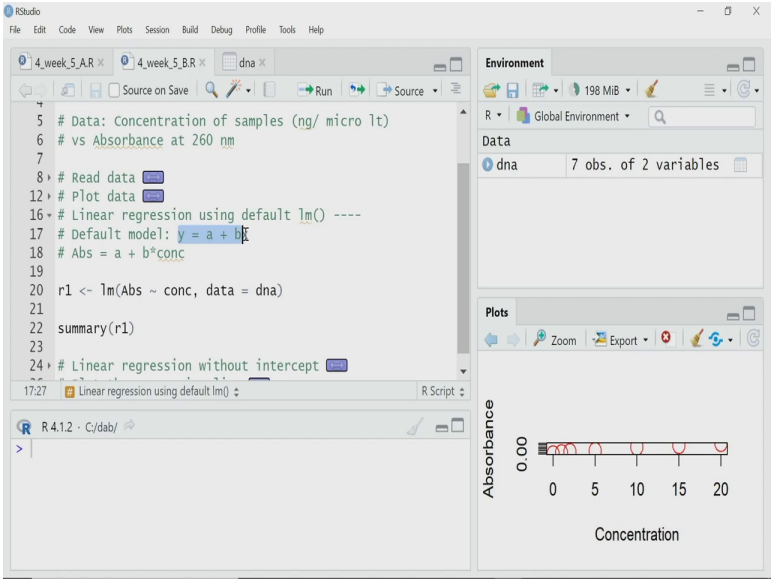
So, once I have that data, now, I may want to plot that data and see how does it look like. So, again, I will use the plot function and I will define the variables in the horizontal axis I want the concentration so it is DNA, the dollar sign and concentration conc, and then the vertical axis I have the absorbance. I want to label x and y axes by concentration and absorbance respectively, I want to use the red colour, and I want to use a circle and other things.

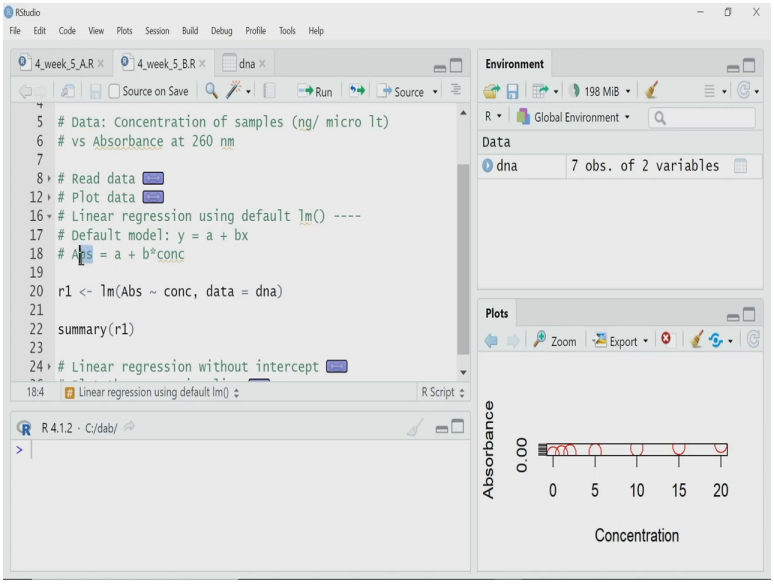
(Refer Slide Time: 17:40)



So, if I execute this, I will get a plot. Let me zoom it to see how does it look like. This is usually a decent plot for DNA absorbance that we do to create the standard plot. Now, I have to fit a straight line to this.

(Refer Slide Time: 17:52)



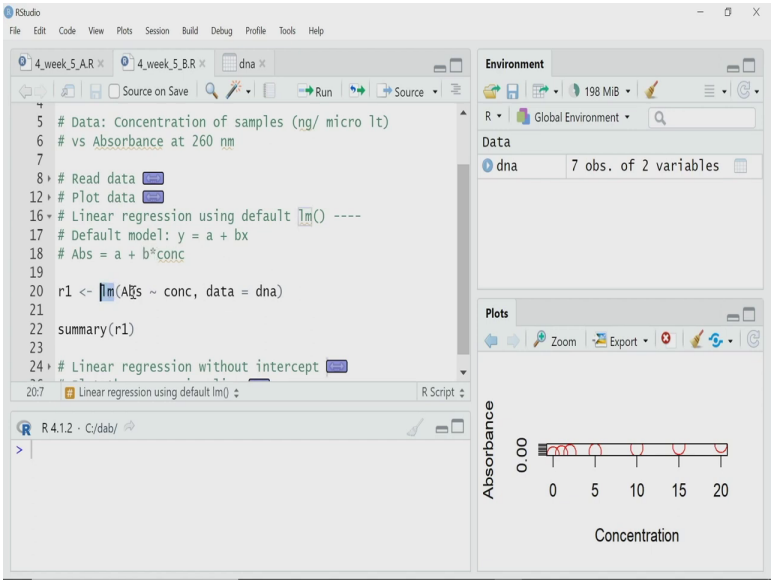


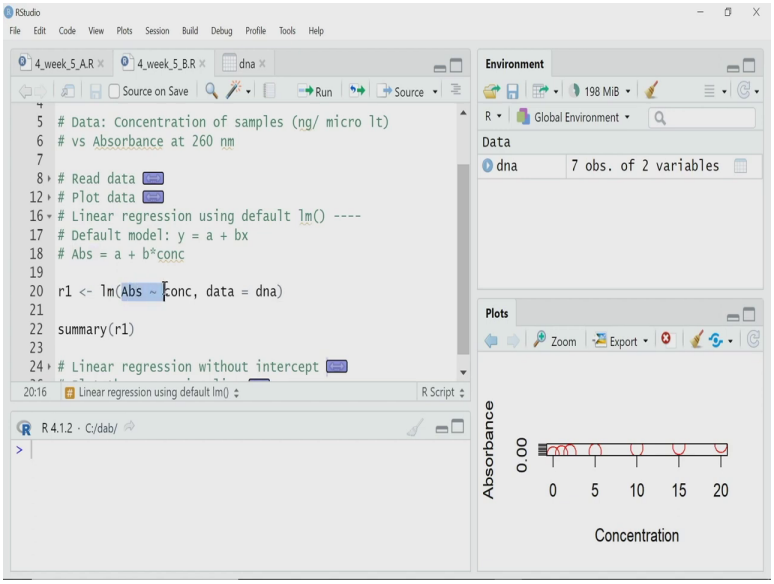
rl← lm(Abs ~ conc, data = dna)

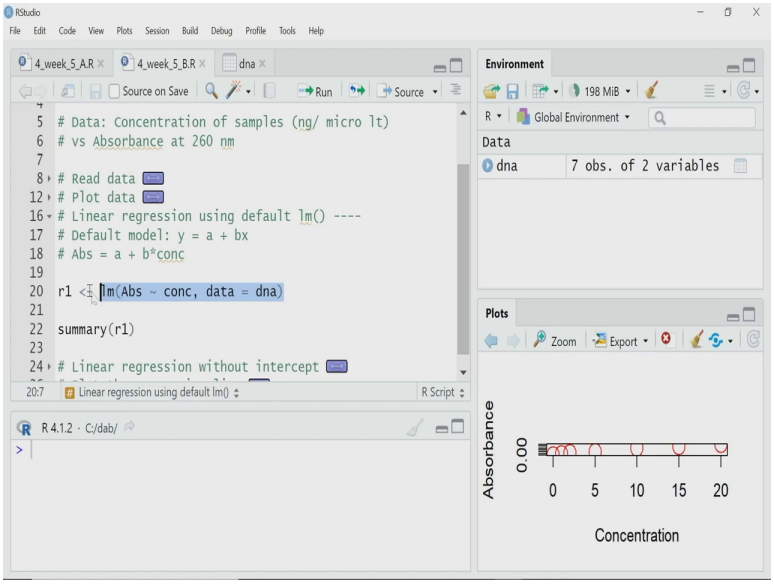
summary(rl)

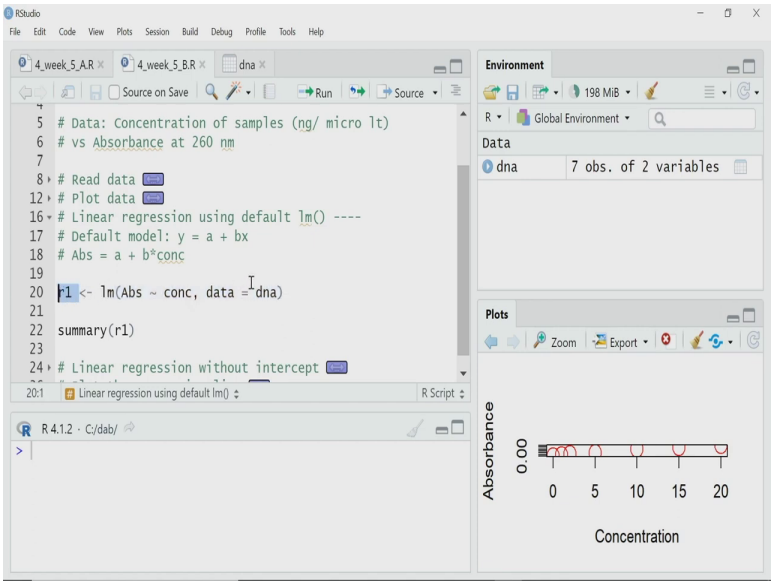
So, to fit a straight line, that means I will do linear regression, simple linear regression. So, I will use the lm function and by lm function by default, we fit the equation y equal to a plus bx and we estimate the coefficient b and a. So, in this case, my model is absorbance equal to a the intercept or coefficient plus b, another coefficient into concentration.

(Refer Slide Time: 18:19)







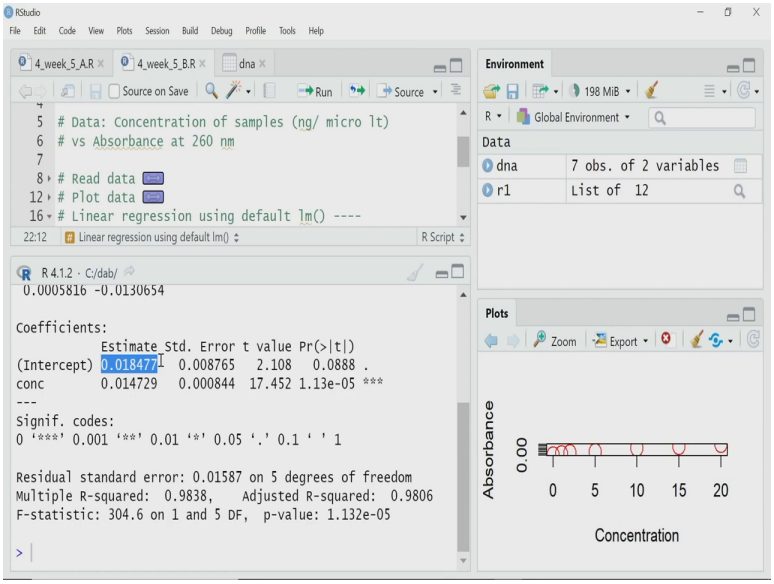


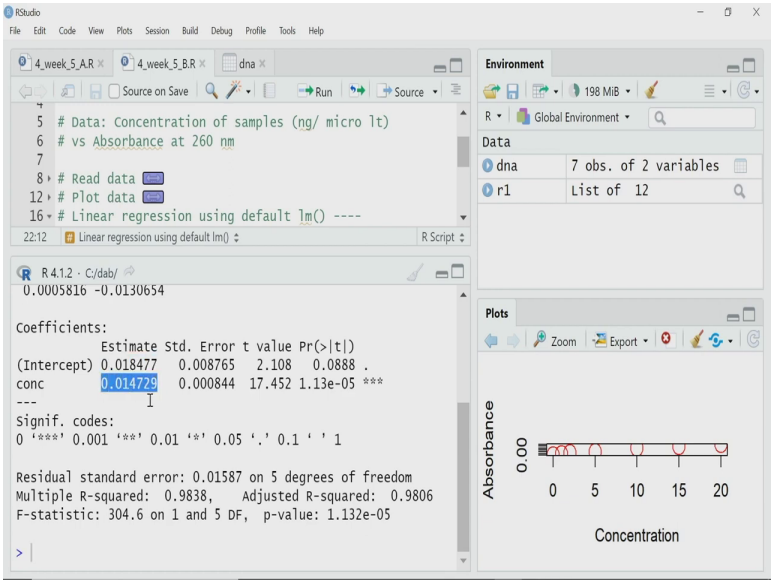
r1← lm(Abs ~ conc, data = dna)

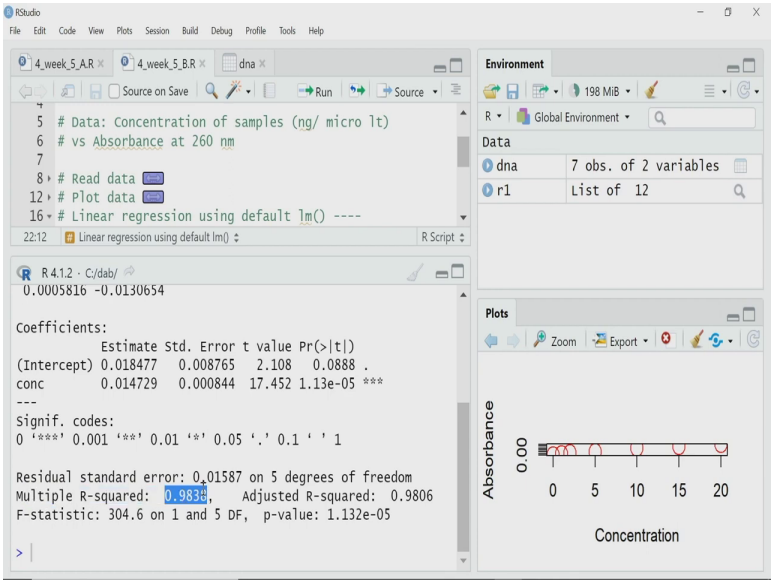
summary(r1)

So, I am calling the lm function. And I am telling it that create a linear model between absorbance and concentration, where concentration is the predictor or the independent variable, and absorbance is the response or dependent variable and use the data equal to DNA and then you calculate the all these things and store that data in the variable r1 regression data in r1. So, here I do the regression. r1 is now storing all my data coming from the regression analysis. Now, I will take a look at the summary of that.

(Refer Slide Time: 19:00)

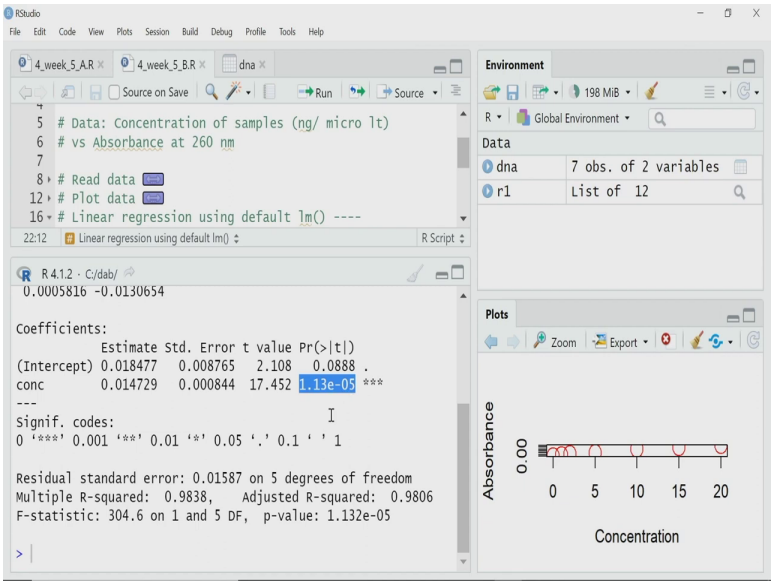


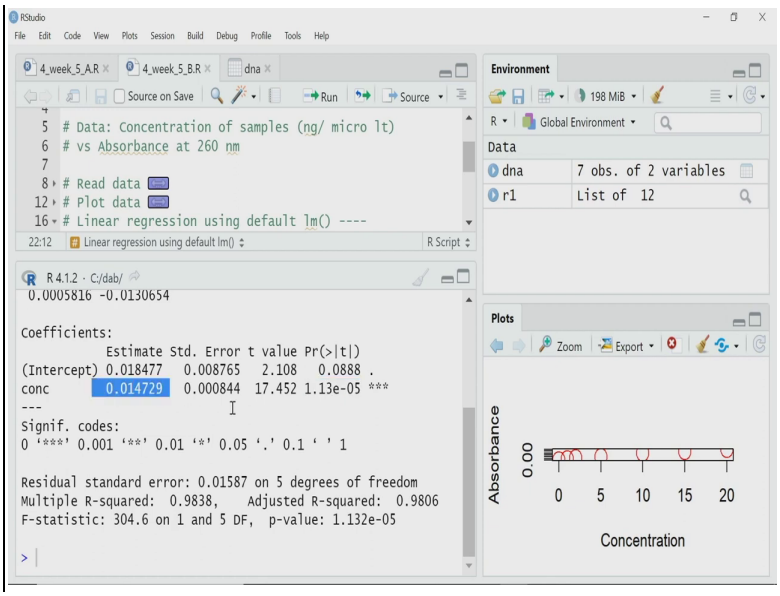




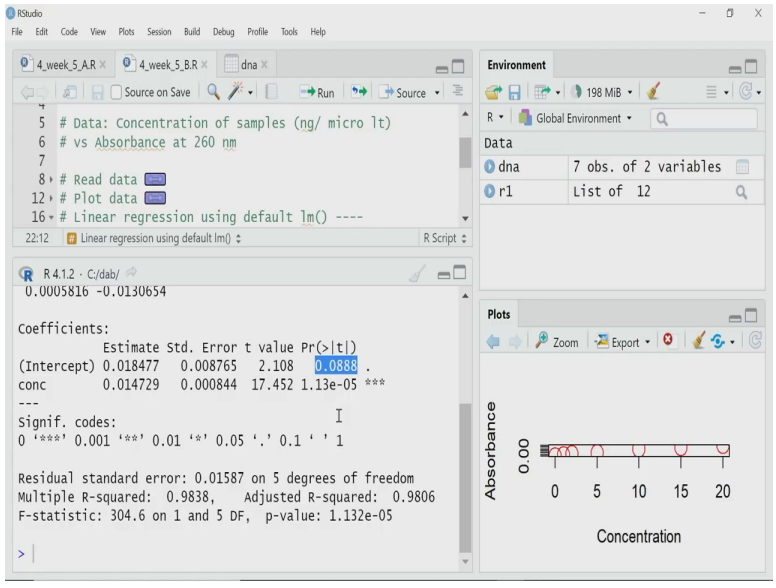
So, if I look into the summary, I will do, what I will get I will get the value of the coefficient, the intercept and the coefficient b and also it will perform t test it will calculate R square everything. So, the intercept is 0.018477 whereas the coefficient for concentration is 0.0147 and R squared value we have to take the multiple R squared just as I said in the previous example, and it is quite good 0.9838.

(Refer Slide Time: 19:31)

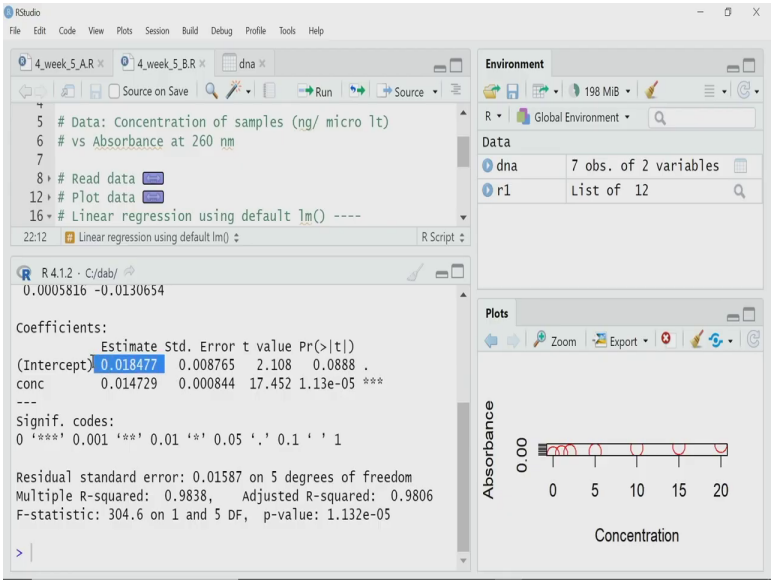



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
dna
Source on Save
Run
Source
5 # Data: Concentration of samples (ng/ micro lt)
6 # vs Absorbance at 260 nm
7
8 # Read data
12 # Plot data
16 # Linear regression using default lm() ----
22:12
Linear regression using default lm()
R Script
R 4.1.2 · C:/dab/
0.0005816 -0.0130654
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.018477 0.008765 2.108 0.0888 .
conc 0.014729 0.000844 17.452 1.13e-05 ***

Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.01587 on 5 degrees of freedom
Multiple R-squared: 0.9838, Adjusted R-squared: 0.9806
F-statistic: 304.6 on 1 and 5 DF, p-value: 1.132e-05
>
Environment
198 MiB
R
Global Environment
Data
dna 7 obs. of 2 variables
r1 List of 12
Plots
Zoom
Export
Absorbance
0.00
0
5
10
15
20
Concentration


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
4_week_5_A.R
4_week_5_B.R
dna
Source on Save
Run
Source
5 # Data: Concentration of samples (ng/ micro lt)
6 # vs Absorbance at 260 nm
7
8 # Read data
12 # Plot data
16 # Linear regression using default lm() ----
22:12
Linear regression using default lm()
R Script
R 4.1.2 · C:/dab/
0.0005816 -0.0130654
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.018477 0.008765 2.108 0.0888 .
conc 0.014729 0.000844 17.452 1.13e-05 ***

Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.01587 on 5 degrees of freedom
Multiple R-squared: 0.9838, Adjusted R-squared: 0.9806
F-statistic: 304.6 on 1 and 5 DF, p-value: 1.132e-05
>
Environment
198 MiB
R
Global Environment
Data
dna 7 obs. of 2 variables
r1 List of 12
Plots
Zoom
Export
Absorbance
0.00
0
5
10
15
20
Concentration

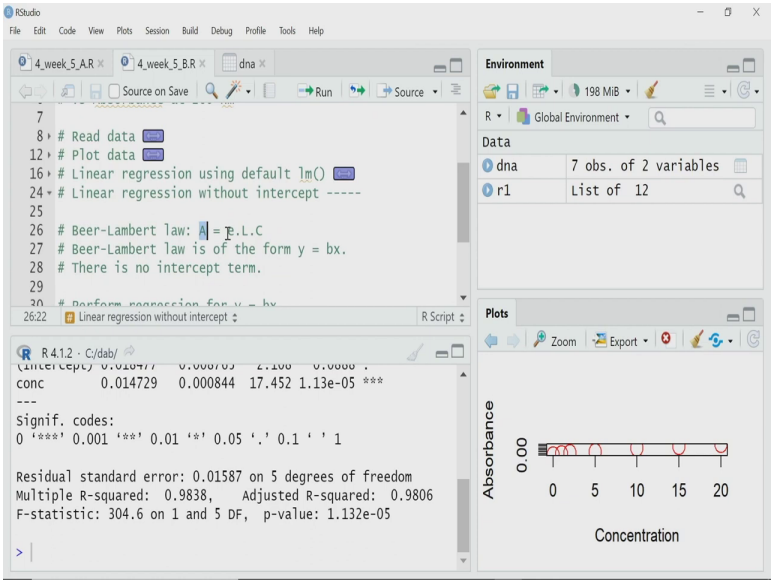


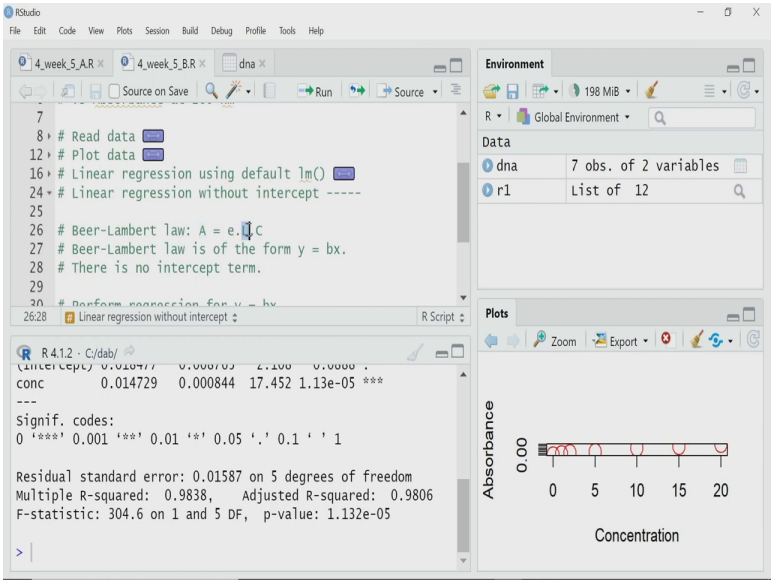
Now, let me look into the t test result for concentration the p value is 1.13 into 10 to the power of minus 5 quite a small value of p that means the coefficient calculated here is statistically significant. But check the p value for the intercept it is 0.088. That means even if I consider a cut up of 0.05 then also it is bigger than that value.

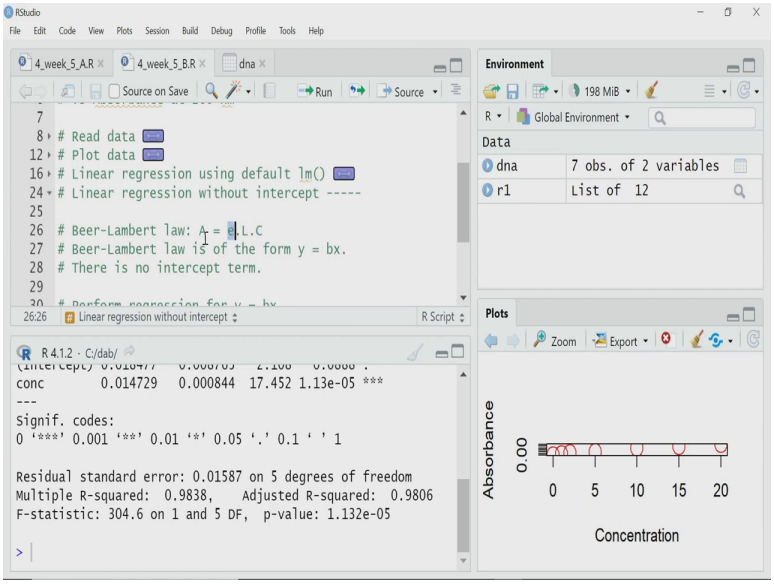
That means even at the level of 0.05 the intercept that I have estimated is not statistically significant. And as I discussed in the lecture on linear regression that means, in from our linear model we should remove this intercept. But there is another way of looking into this problem, which is more connected with what the experiment I have performed. Let me explain that.

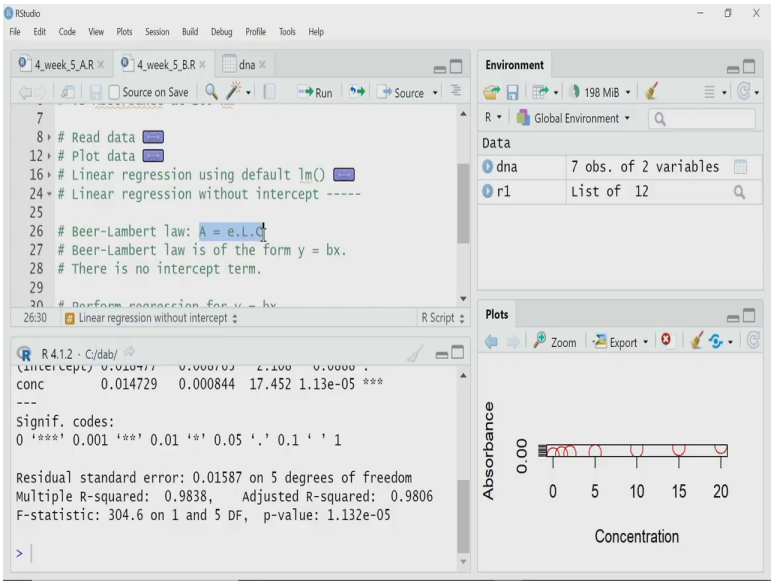
So, what experiment I have performed, I have performed an experiment where I have taken different samples of the same DNA with different concentration I have measured the absorbance. And we know if I can make a sufficiently diluted solution of this DNA, these result should follow Beer-Lambert law and I hope you remember the Beer-Lambert law.

(Refer Slide Time: 20:52)







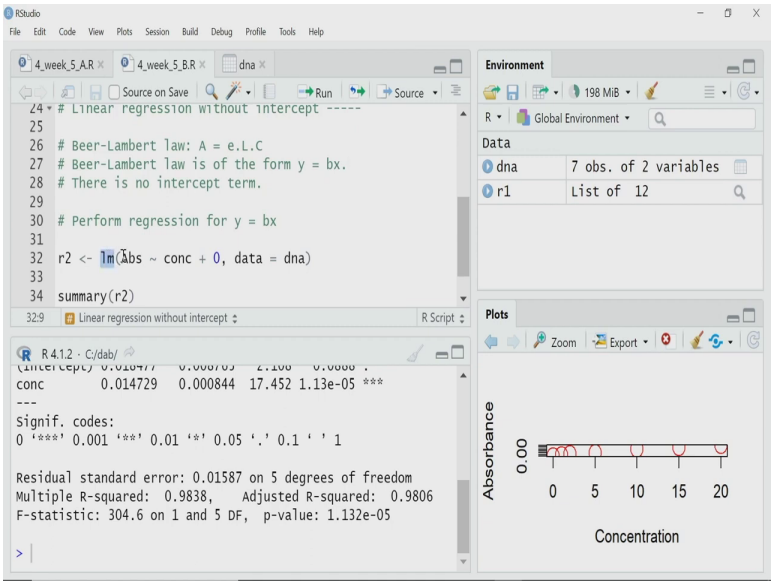


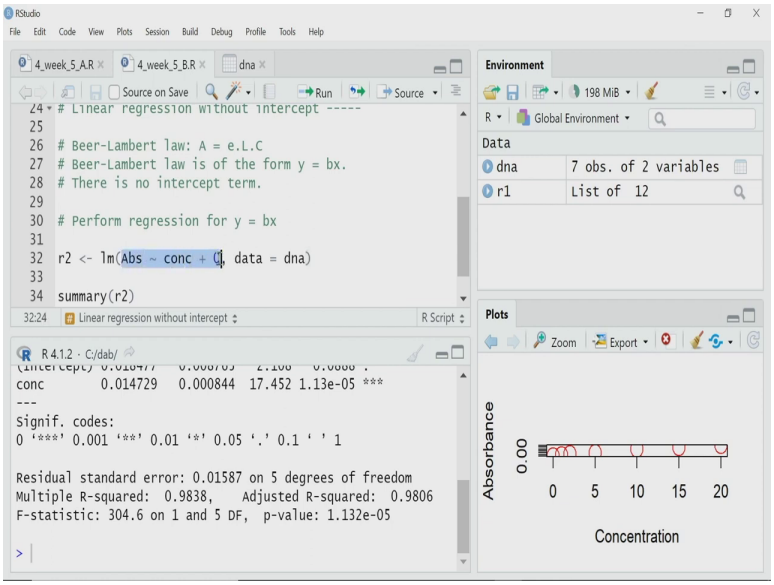
Just to remind you what is there in Beer-Lambert law, in beer Lambert law the absorbance of the sample should be equal to L, L is the path length into C the concentration and in the beginning I have a constant term. So, A call to eta into L into C. Now, that means A and C is linearly connected, but remember there is no intercept term here. So, the equation equivalent to this is y equal to bx not y equal to bx plus a.

So, Beer-Lambert law itself does not have an intercept term. Now, my whole experiment is relying upon actually Beer-Lambert law. That is why I believe concentration and absorbance should have a linear relation. And the law itself says that there is no intercept term. That means I should never use y equal to bx plus a form of linear regression in this case. I have to perform

linear regression without the intercept, I will tell my lm function that see there is no intercept here it is y equal to bx. So, that I can actually specify when I am using the lm function. How can I do that?

(Refer Slide Time: 22:03)





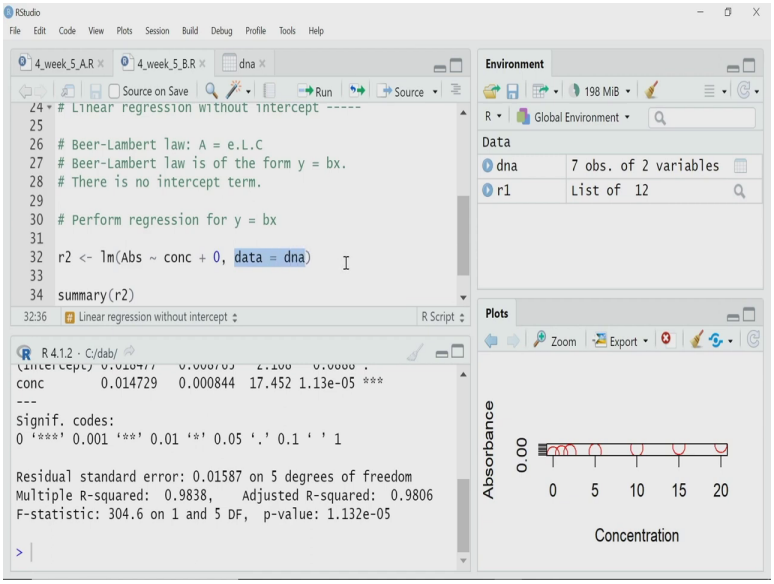
r2 ← lm(Abs ~ conc + 0, data = dna)

summary(r2)

That is what I have shown here. I am calling the lm function and I am telling the to lm function, that model is absorbance and concentration plus 0, that means this by plus 0 writing this plus 0, I

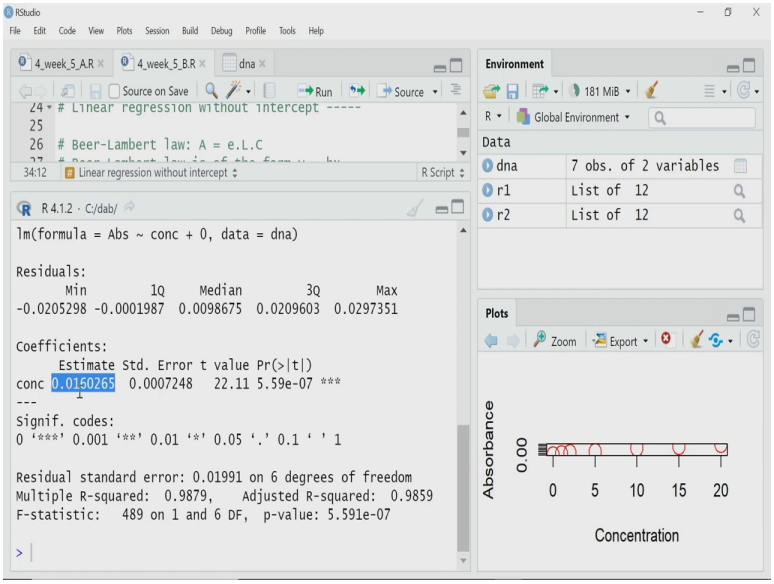
am specifying the function that see in this case you have to set intercept equal to 0, do not estimate the value of intercept.

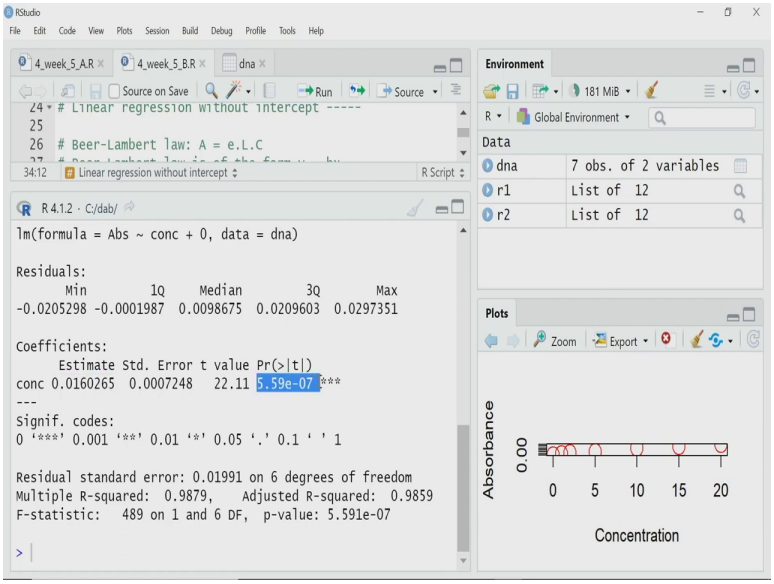
(Refer Slide Time: 22:26)

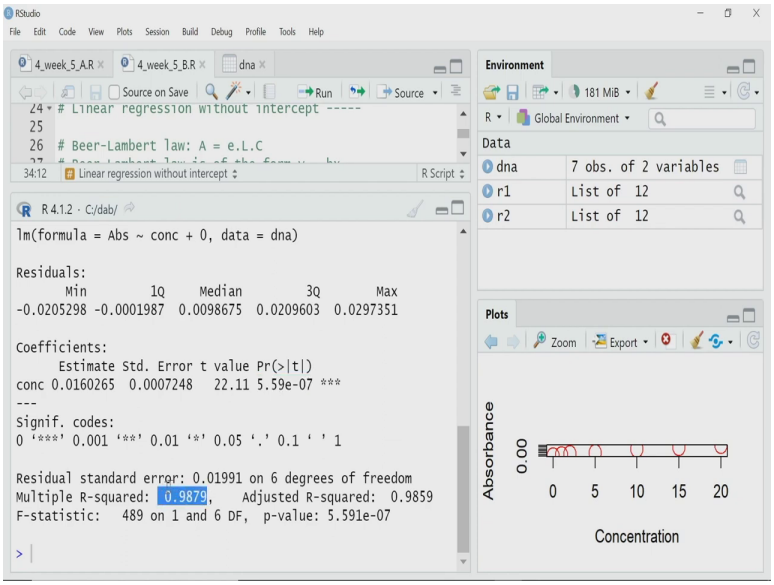


And then obviously, the data is equal to DNA on which I want to perform the regression. I performed it and now let me check the summary of this by calling the summary function.

(Refer Slide Time: 22:38)

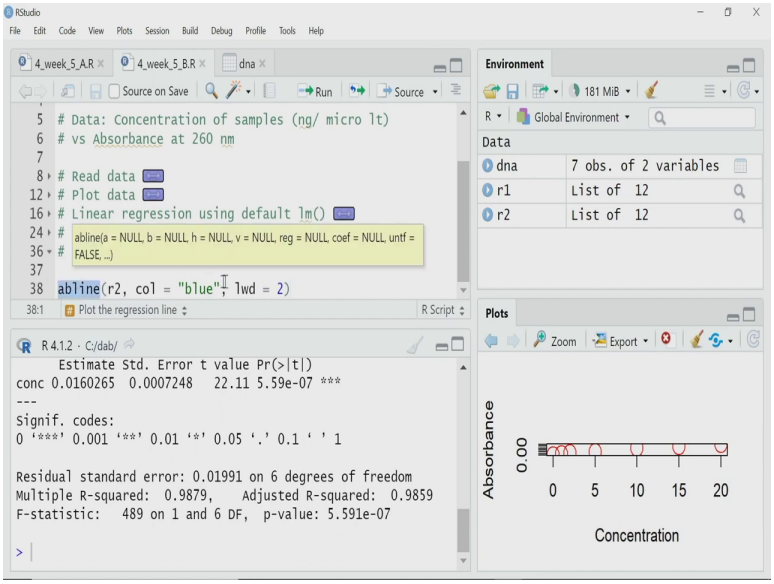


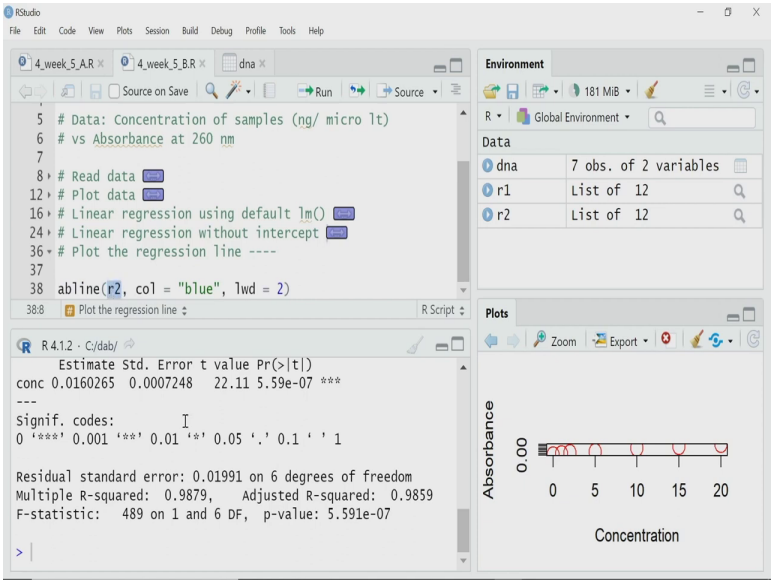


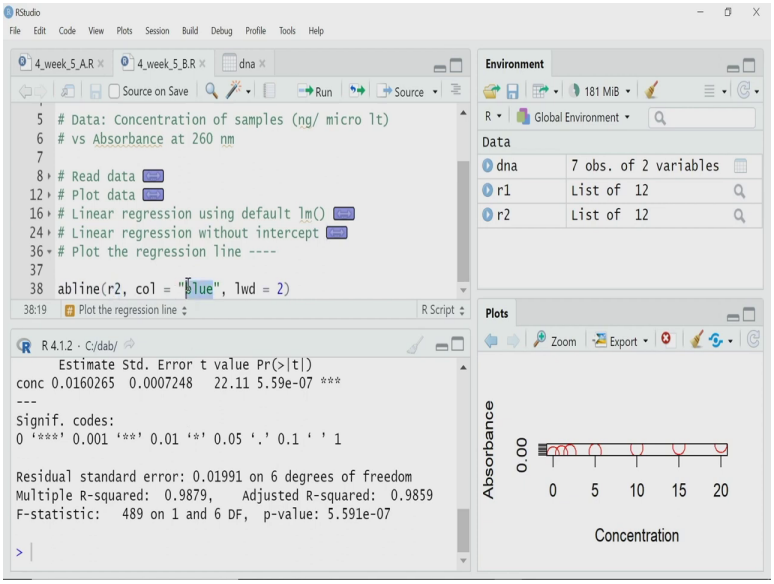


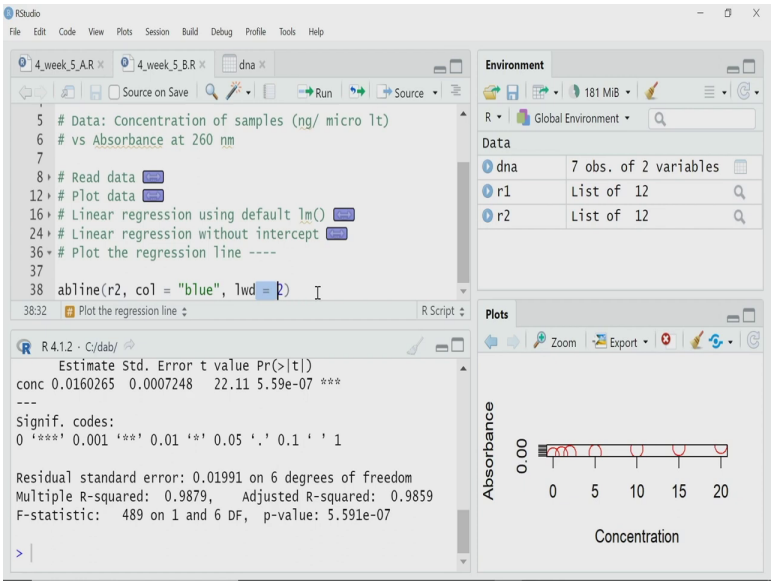
And now I can check the values of the coefficients. Now, remember, in this case, that there is no intercept. So, only one coefficient is calculated that is for the concentration and that is given here 0.016 and its p value is very low, that means its significant, I have the R squared value and that is quite decent enough. So, that means now I have performed the regression properly. Now, I have created a linear model for my concentration versus absorbance data which follow Beer Lambert law correctly. So, I have performed it.

(Refer Slide Time: 23:24)





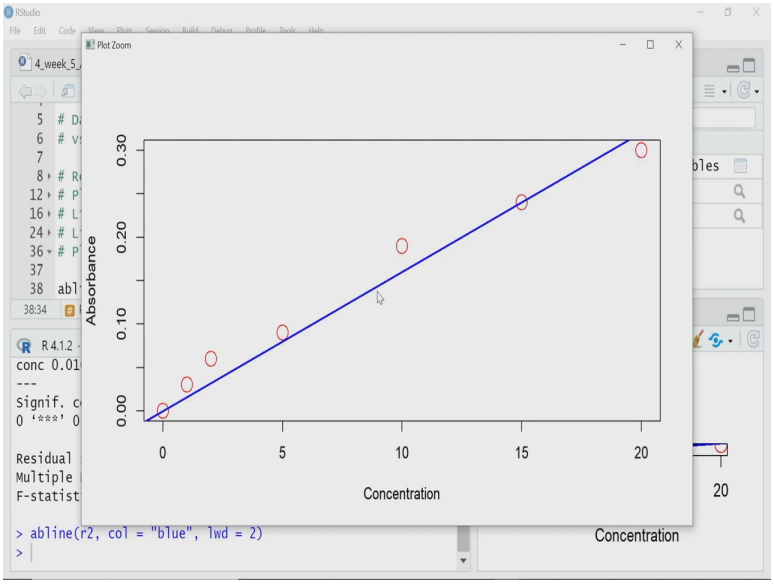




abline(r2, col= ”blue”, lwd = 2)

Now, what I will do, I will plot this regression line, so that I can use that line to calculate the concentration of an unknown sample. So, to do that, I will draw the line using the abline function, just I have done in the other case, in this case, r2 is the regression data set and I want a blue colour with line thickness 2.

(Refer Slide Time: 23:37)



So, here you can easily see, this is my red circles are my data and this blue thick line is my regressed line that is my standard curve. And just to remind you again, when we use UV visible spectroscopy, when we are using sufficiently diluted solutions, we believe that this experiment follow Beer-Lambert law. In Beer-Lambert law absorbance and concentration has a linear relationship, but in that linear relationship, there is no intercept term.

So, that means when I will perform linear regression, I should always remove that intercept term. And you can easily understand that if I do not have any intercept, when 0 is the concentration absorbance should be 0. And that is what you have done when you have set the machine at auto 0. That is all for this lecture on linear regression, we have taken two example and in both cases, we have used the lm function to perform the linear regression and then overlay the regressed line on the original data set. That is all for this lecture. Thank you for learning with me today.