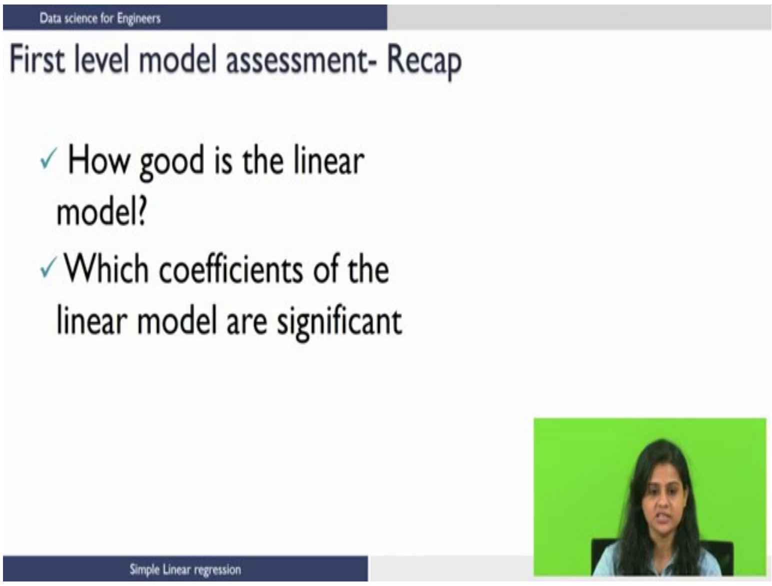
Data Science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture-37
Simple Linear Regression Model Assessment

(Refer Slide Time: 00:21)



Data science for Engineers

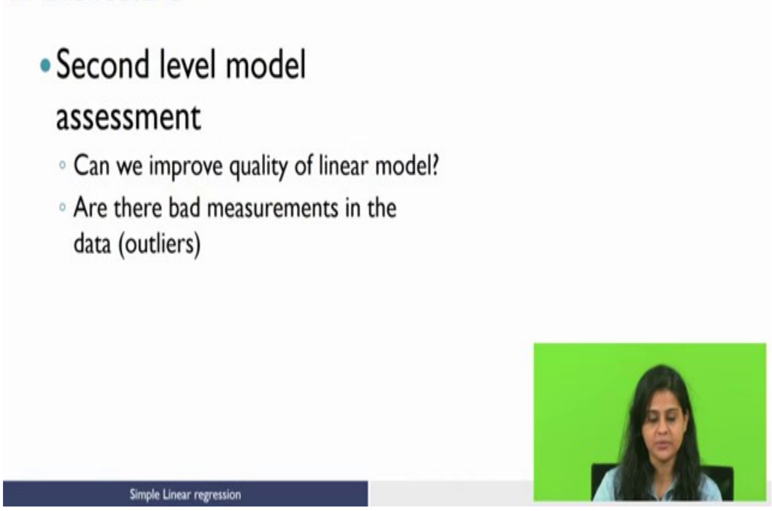
First level model assessment- Recap

- ✓ How good is the linear model?
- ✓ Which coefficients of the linear model are significant

Simple Linear regression

Welcome to the third lecture on implementation of simple linear regression using R. In the last lecture, we looked at the first level of model assessment, we saw how good is a linear model that we built and we also saw, how to identify the significant coefficients in the linear model.

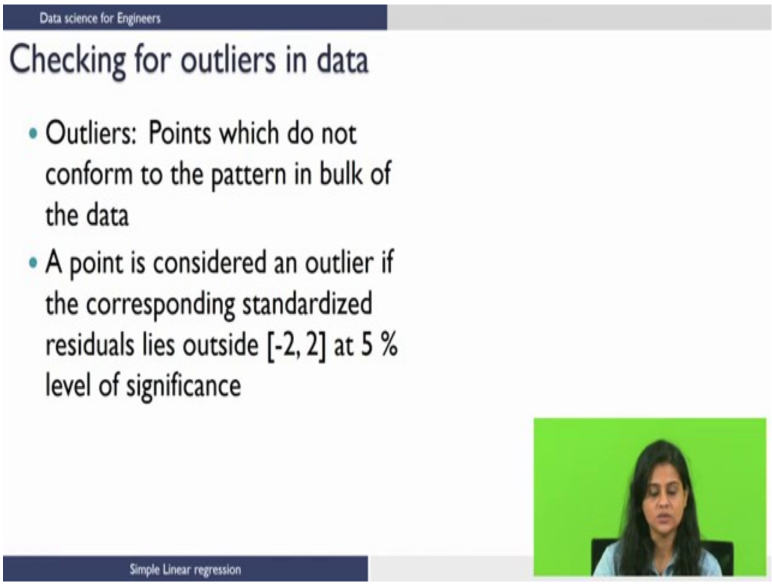
(Refer Slide Time: 00:35)



- Second level model assessment
  - Can we improve quality of linear model?
  - Are there bad measurements in the data (outliers)

In this lecture, we are going to look at the second level of model assessment. As a part of this, we are going to see, if we can improvise the quality of the linear model and can we identify bad measurements and by bad measurements, we mean outliers.

(Refer Slide Time: 00:52)

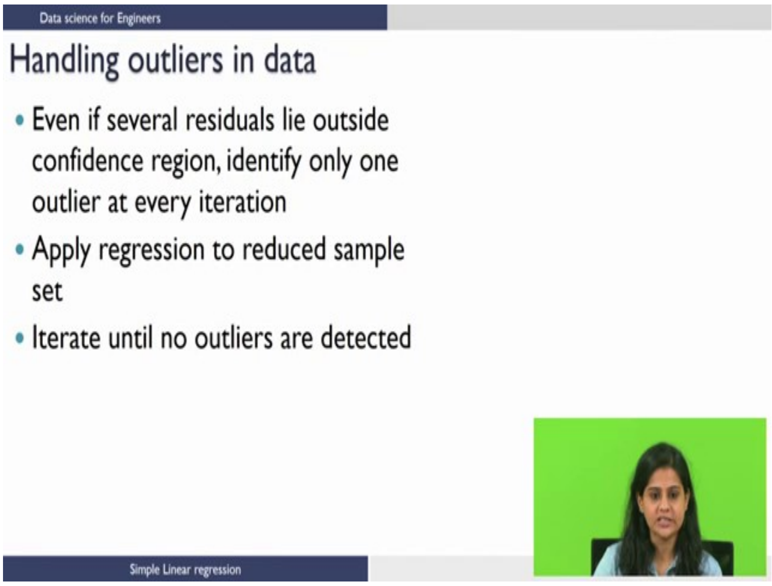


Checking for outliers in data

- Outliers: Points which do not conform to the pattern in bulk of the data
- A point is considered an outlier if the corresponding standardized residuals lies outside [-2, 2] at 5 % level of significance

So, let us see, what outliers are. So, outliers are points, which do not con-form to the bulk of the data. Now, a point is considered an outlier, if the corresponding standardized residual falls outside, - 2 and + 2 at 5 per-cent significance level.

(Refer Slide Time: 01:10)



Data science for Engineers

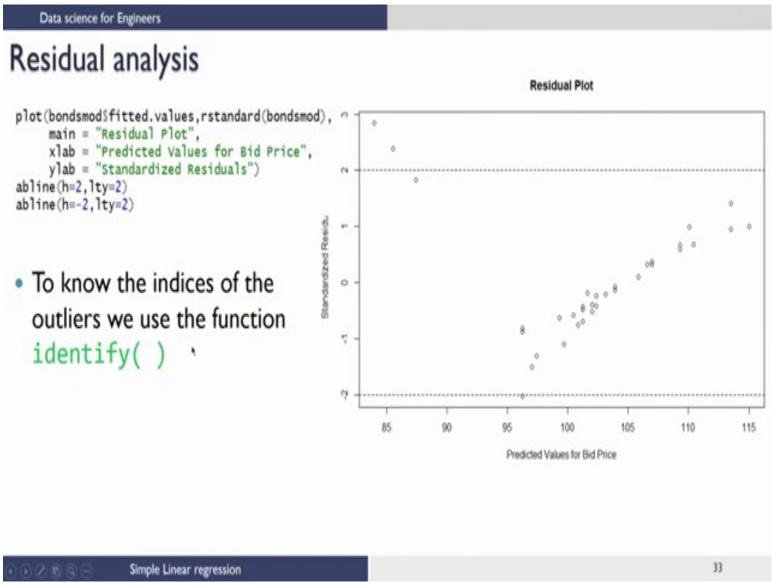
Handling outliers in data

- Even if several residuals lie outside confidence region, identify only one outlier at every iteration
- Apply regression to reduced sample set
- Iterate until no outliers are detected

Simple Linear regression

Now, let us see how to handle these outliers, even if we have several outliers which lie outside the confidence region, we are going to identify only one at a time, at every iteration and after doing so, we are going to apply a linear model on the reduced sample. Now, we are going to iterate, till we detect no more outliers. Now, let us see how to handle these outliers. We are going to start with the residual analysis.

(Refer Slide Time: 01:36)



Data science for Engineers

Residual analysis

```
plot(bondsmod$fitted.values,rstandard(bondsmod),
     main = "Residual Plot",
     xlab = "Predicted Values for Bid Price",
     ylab = "Standardized Residuals")
abline(h=2,lty=2)
abline(h=-2,lty=2)
```

- To know the indices of the outliers we use the function `identify( )`

Simple Linear regression

33

So, you can see a plot function on the left hand side. Now, for the residual plot, I am going to plot fitted values on the x axis and the standardized residual from the model we have built. Now, we built a linear model called bondsmod and we are going to calculate the standardized residuals for it.

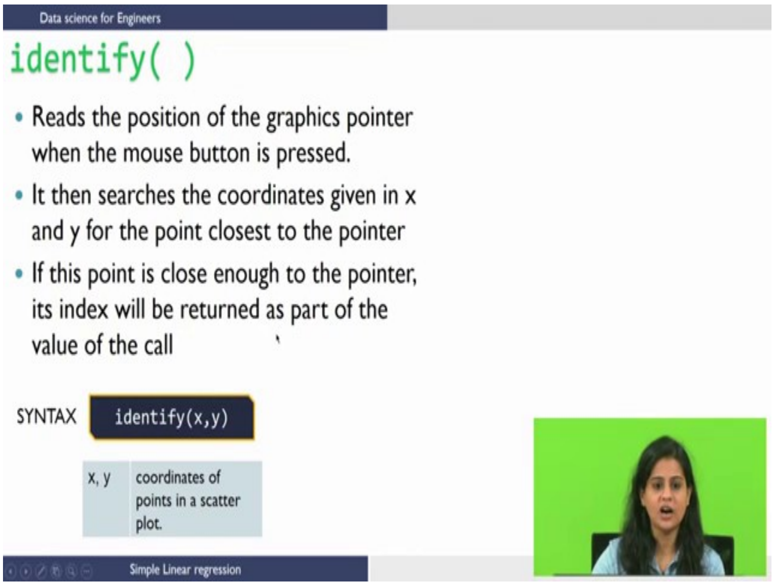
So, that becomes my y. I am also giving a title like I earlier said. The title for this plot is residual plot and my x label is nothing, but predicted values for bid price. Similarly, my y label is standardized residual. Now, after doing so, we need to set the confidence region. So, let us see how to do that. We again use the same command a b line. Now, a b line is what we have used to fit the linear model onto the plot.

Now, with the same command, we can give the confidence region as well. Now, I am going to set the height, which = h here, as 2 and the line type as 2. So, height is at which you want the line to be drawn and line type is nothing but how you want the line to be drawn. So, you can have dashed lines solid line, dashed and a dot. So, you have several options in there similarly, I also need a lower confidence limit. So, I am setting that to be = - 2 and for the same limit I am setting the line type to be = 2.

Now, let us see how the plot looks. On the right hand side, I have the plot. So, we can see that there are two lines drawn at + and - 2 that defines the confidence level. Now, on the y axis, I have standardized residuals and on the x axis, I have predicted values for bid price. Now, from the plot, we can see that there are two outliers, which are really farther, there is one, which is close to the upper confidence limit. And there is one, which is exactly almost close to the lower confidence limit.

So, let us see how to identify these. So, from the plot, we may not be able to tell which points are these. By points, I mean in the row IDs. We are going to use another function called identify, that will help us identify the indices of these samples. Now, let us see what identify function does.

(Refer Slide Time: 03:53)



Data science for Engineers

identify( )

- Reads the position of the graphics pointer when the mouse button is pressed.
- It then searches the coordinates given in x and y for the point closest to the pointer
- If this point is close enough to the pointer, its index will be returned as part of the value of the call

SYNTAX `identify(x,y)`

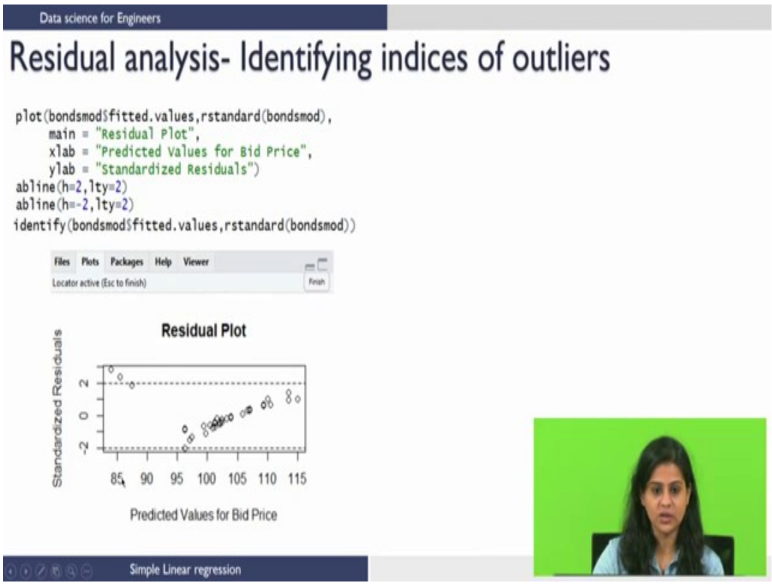
x, y	coordinates of points in a scatter plot.

Simple Linear regression

So, it treats the position of the graphic pointer, when the mouse button is pressed it, then searches the coordinates given an x and y for the point closest to the pointer. Now, if the point is close enough to the pointer, its index will be returned as a part of the value code.

Now, let us look at the syntax for it. So, identify is a function and x and y are my input parameters. So, what are my x and y, they are the coordinates of the points in the scatter plot. Now, let us see how to use this function to identify the indices.

(Refer Slide Time: 04:27)



Data science for Engineers

Residual analysis- Identifying indices of outliers

```
plot(bondsmod$fitted.values,rstandard(bondsmod),
     main = "Residual Plot",
     xlab = "Predicted Values for Bid Price",
     ylab = "Standardized Residuals")
abline(h=2,lty=2)
abline(h=-2,lty=2)
identify(bondsmod$fitted.values,rstandard(bondsmod))
```

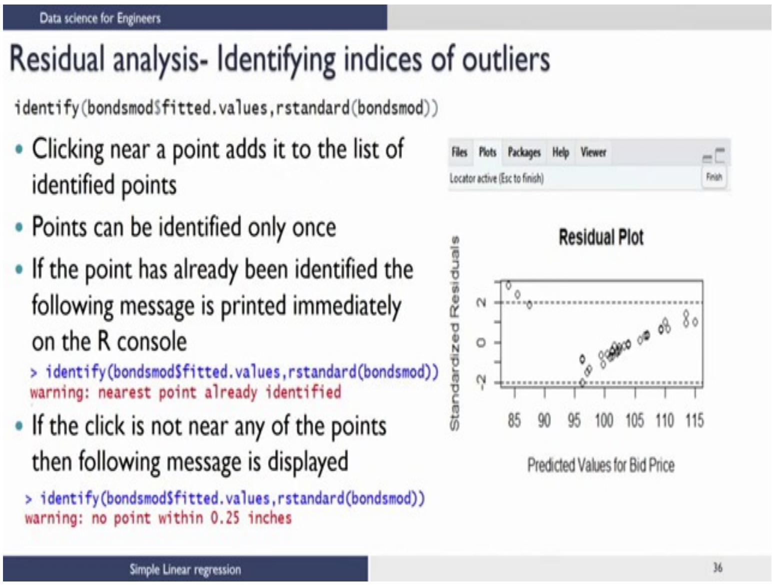
Locator active (Esc to finish)

Simple Linear regression

On my left, I have the same commands for the residual plot. So, this is what we saw in the last slide. I now, use the identify function to identify the indices. Now, my input for this is fitted values of the bonds model and the standardized residuals the reason. I am giving fitted values is, because on the plot, I want the indices to be found. So, the plot has fitted values from the model and the standardized residuals from the model.

So, on this plot I want my indices to be identified. So, I give the same inputs that I have used for the plot command for identify function. So, again here, if you see I have fitted values from the bonds model and I am plotting for the y parameter, I am plotting the standardized residuals. Now, once you execute the command, you will not get the output immediately. What will be displayed is the following snippet on the left, you will see a finish button and you will see a message being displayed. Now, on this plot, we will need to click and identify each of the points. Now, let us see how to do that.

(Refer Slide Time: 05:33)



Residual analysis- Identifying indices of outliers

```
identify(bondsmod$fitted.values,rstandard(bondsmod))
```

- Clicking near a point adds it to the list of identified points
- Points can be identified only once
- If the point has already been identified the following message is printed immediately on the R console

```
> identify(bondsmod$fitted.values,rstandard(bondsmod))
warning: nearest point already identified
```

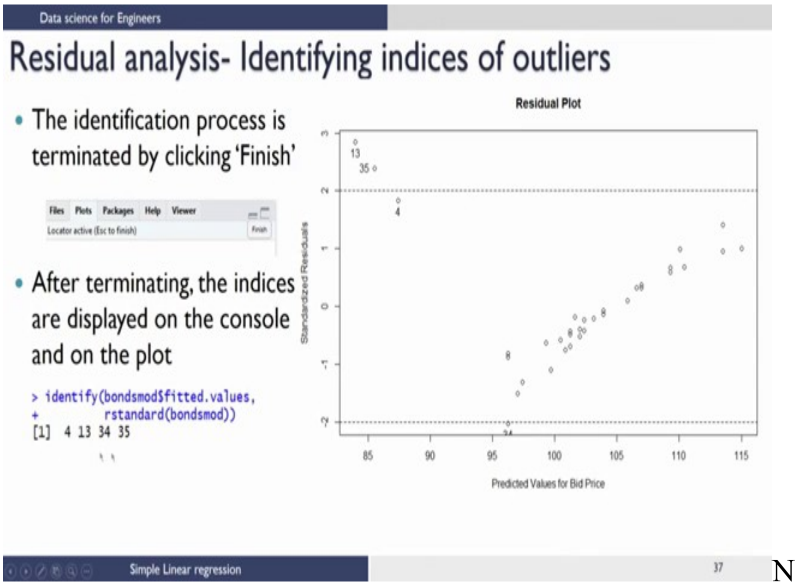
- If the click is not near any of the points then following message is displayed

```
> identify(bondsmod$fitted.values,rstandard(bondsmod))
warning: no point within 0.25 inches
```

So, I am displaying the command above to remind, you of the fact that we are using fitted values and standardized residuals to identify. Now, click it near a point, adds it to the list of the identified points. Now, if I am going to click near this point. It is going to identify this point and store it. Now, all these points can be identified only once. Now, if a point has already been identified and you still click near it, then you will get the following message. It will be a warning, which reads as nearest point already identified.

Now, if you do not click near any of the points, then a message is displayed, which says that no point is identified within 0.25 inches. So, if I click here, then I do not have any points closest to it. So, it will display a message saying no point within 0.25 inches.

(Refer Slide Time: 06:29)



Data science for Engineers

Residual analysis- Identifying indices of outliers

- The identification process is terminated by clicking 'Finish'
- After terminating, the indices are displayed on the console and on the plot

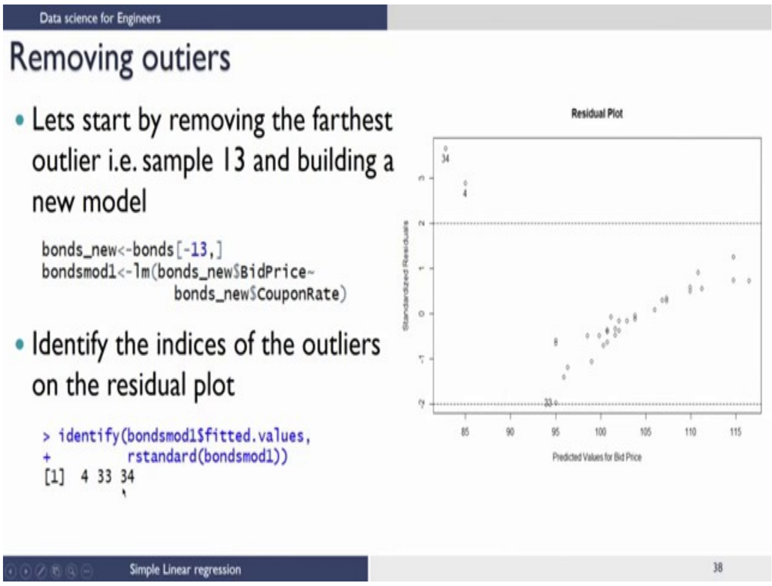
```
> identify(bondsmod$fitted.values,
+          rstandard(bondsmod))
[1]  4 13 34 35
```

Simple Linear regression 37

Now, once you have identified all the outliers, you need to click the finish button that is present on the top right, corner of the graphical window, you can also press escape to finish. Now, after terminating the indices are displayed on the console and on the plot. Now So, you can see on the console, I have the indices being displayed as 4 13 34 35, but this will give you only the value.

So, now, to know where your outliers lie on the plot, I am going to look at the plot. So now, I know the 13th point of the sample is the farthest, which is here, followed by the 35th sample, followed by the sample 4 and then I have one more sample, which is here, which is the 34th sample. So, after identifying these outliers, we are going to start by removing one at a time and we are going to build a new model. Now, let us see how to do that. Now, I will start by removing one point at a time.

(Refer Slide Time: 07:32)



Data science for Engineers

Removing outiers

- Lets start by removing the farthest outlier i.e. sample 13 and building a new model

```
bonds_new<-bonds[-13,]
bondsmod1<-lm(bonds_new$BidPrice~
               bonds_new$CouponRate)
```

- Identify the indices of the outliers on the residual plot

```
> identify(bondsmod1$fitted.values,
+          rstandard(bondsmod1))
[1]  4 33 34
```

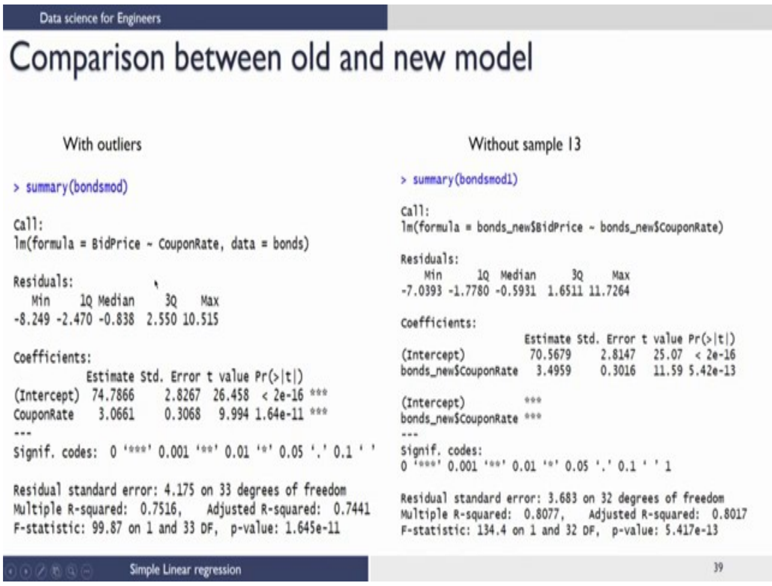
Simple Linear regression 38

The first point that I am going to remove is sample 13 that is the 13th point, because it is the farthest in the plot. So, to start with, I am going to create a new data frame called bonds new and it will have all rows of bonds except the 13th row. So, then I am going to create another object called bonds mod one, which is the linear model that is being built for the new data. So, I am going to regress bid price from the new data frame bonds new with coupon rate from the same data set. Now, after building the new linear model, which does not contain the 13th point, that is an outlier.

We are going to repeat the same process again that is on the residual plot, we are going to identify the outliers for the new data. So, on my, right. I already have the residual plot with the outliers being identified. So, from the snippet, we can see that for the new data, I have my 4th point, 33rd point and 34th point being, are being identified as outliers. So, now, this new data will contain only 34 data observations, because we have already removed one observation. So, the indices for the new data will change.

So, the farthest point in this data is the 34th point and after that I have the 4th point and followed by that, there is also one point on the line, which is the 33rd point, for this new data. Now, we can see that, if you compare this plot and the earlier plot this point, which is located here was below this line and that is because we had an extreme outlier in the previous case, that had a smearing effect on the remaining points. Now, after building this new linear model let us take a look at the summary.

(Refer Slide Time: 09:21)



Data science for Engineers

Comparison between old and new model

With outliers

```
> summary(bondsmod)

Call:
lm(formula = BidPrice ~ CouponRate, data = bonds)

Residuals:
   Min     1Q Median     3Q    Max 
-8.249 -2.470 -0.838  2.550 10.515 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  74.7866     2.8267  26.458  < 2e-16 ***
CouponRate    3.0661     0.3068   9.994 1.64e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Residual standard error: 4.175 on 33 degrees of freedom
Multiple R-squared:  0.7516,	Adjusted R-squared:  0.7441 
F-statistic: 99.87 on 1 and 33 DF,  p-value: 1.645e-11
```

Without sample 13

```
> summary(bondsmod1)

Call:
lm(formula = bonds_new$BidPrice ~ bonds_new$CouponRate)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.0393 -1.7780 -0.5931  1.6511 11.7264 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           70.5679     2.8147   25.07  < 2e-16
bonds_new$CouponRate   3.4959     0.3016   11.59 5.42e-13
                        
(Intercept)          ***
bonds_new$CouponRate ***
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.683 on 32 degrees of freedom
Multiple R-squared:  0.8077,	Adjusted R-squared:  0.8017 
F-statistic: 134.4 on 1 and 32 DF,  p-value: 5.417e-13
```

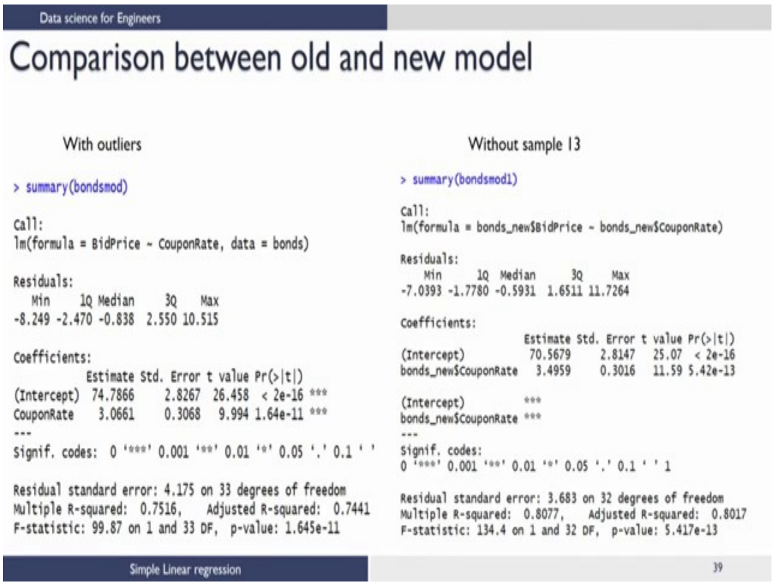
Simple Linear regression

39

.

On the left is the summary of the old model bondsmod that contains all the points. On the right, I have the summary of the new model, which does not contain the 13th sample. So, from the R squared values of the two model, we can see that there is a drastic change by just removing one extreme point. So, from 0.17516, the R square improves to 0.8077. So, that is a quite drastic change. Now, let us remove all the other points one by one and let us see how the R squared value changes.

(Refer Slide Time: 10:00)



Data science for Engineers

Comparison between old and new model

With outliers

```
> summary(bondsmod)

Call:
lm(formula = BidPrice ~ CouponRate, data = bonds)

Residuals:
   Min     1Q Median     3Q    Max 
-8.249 -2.470 -0.838  2.550 10.515 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  74.7866     2.8267  26.458  < 2e-16 ***
CouponRate    3.0661     0.3068   9.994 1.64e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Residual standard error: 4.175 on 33 degrees of freedom
Multiple R-squared:  0.7516,	Adjusted R-squared:  0.7441 
F-statistic: 99.87 on 1 and 33 DF,  p-value: 1.645e-11
```

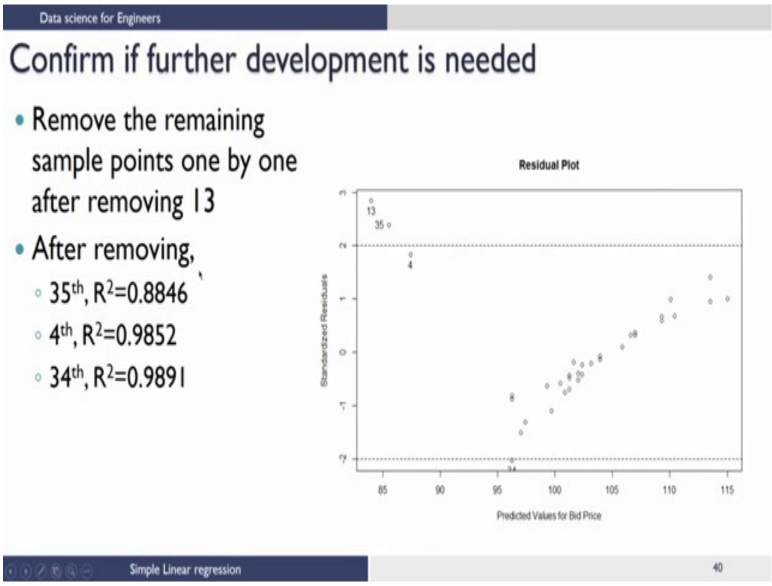
Without sample 13

```
> summary(bondsmod1)

Call:
lm(formula = bonds_new$BidPrice ~ bonds_new$CouponRate)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.0393 -1.7780 -0.5931  1.6511 11.7264 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           70.5679     2.8147   25.07  < 2e-16
bonds_new$CouponRate   3.4959     0.3016   11.59 5.42e-13
                        
(Intercept)          ***
bonds_new$CouponRate ***
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.683 on 32 degrees of freedom
Multiple R-squared:  0.8077,	Adjusted R-squared:  0.8017 
F-statistic: 134.4 on 1 and 32 DF,  p-value: 5.417e-13
```

Simple Linear regression

39

(Refer Slide Time: 10:04)



Data science for Engineers

Confirm if further development is needed

- Remove the remaining sample points one by one after removing 13
- After removing,
  - 35th, R^2=0.8846
  - 4th, R^2=0.9852
  - 34th, R^2=0.9891

Simple Linear regression

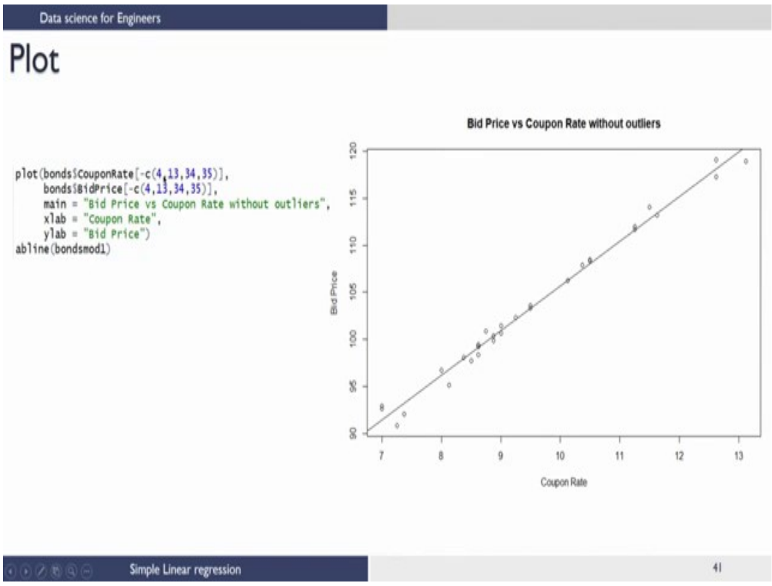
40

Now, I am removing the remaining points one by one. So, earlier I started by removing 13th point. Now, I am going to remove the 35th. So, let us see, what the a square value is, after removing the 35th point. So, the R square changes from 0.80 to 0.88. So, there is a quite big leap here as well.

So, after removing the 35th point, I am able to see a pretty good change in the R squared value. Now, let us look at what the R squared value is If I remove the fourth point. So, the R squared value improves from 0.88 to 0.98. So, that is also pretty good jump. So, now, these indices are for the old data. So, I also have one more index to remove, which is index 34. Now, let us see what happens, if we remove this.

So, after I remove the 34th point my R squared slightly increases; that is also from the 3rd decimal place. So, from 0.9852 it increases to 0.9891. So, the difference is not huge. We need not treat this point as an outlier by itself, because it does not improvise the model any further. So, now, after removing all these four points, we are going to plot the new regression line over the data.

(Refer Slide Time: 11:27)



Data science for Engineers

Plot

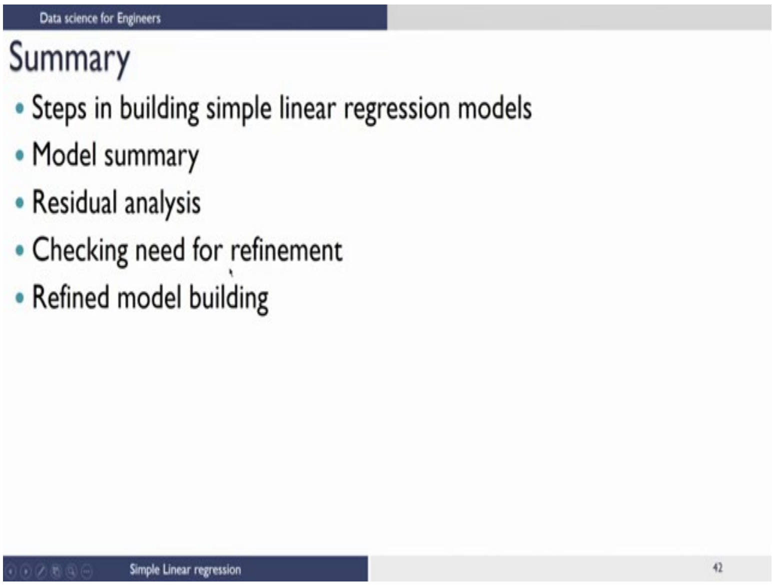
```
plot(bonds$CouponRate[-c(4,13,34,35)],
     bonds$BidPrice[-c(4,13,34,35)],
     main = "Bid Price vs Coupon Rate without outliers",
     xlab = "Coupon Rate",
     ylab = "Bid Price")
abline(bondsmod1)
```

Simple Linear regression

41

So, on the left you can see, that I have removed the 4 index basically, 4 13 34 and 35. These points I have removed from my data and similarly, for bid price also, I have removed these points and I am going to t the new model. So, bondsmod one does not have any outliers now and I am going to plot the regression line over the data. So, our regression line fits the data pretty well, though there are some points, which are really away, but it does not change the nature of the slope drastically. So, this is a pretty good model and we have removed all the possible outliers that we thought were influencing the regression line.

(Refer Slide Time: 12:19)



Data science for Engineers

Summary

- Steps in building simple linear regression models
- Model summary
- Residual analysis
- Checking need for refinement
- Refined model building

Simple Linear regression

42

So, to summarize in this three lectures, we looked at the steps, which are taken in building a simple linear regression model, we saw how to interpret the results from the summary, for these models. We looked at residual analysis. So, we looked at answering some of the question as how to treat outliers, we also saw how to identify significant coefficients in our model and how good our model is. We also saw the need for checking for refinement of existing models and then we built a refined model without any outliers.

Thank you.