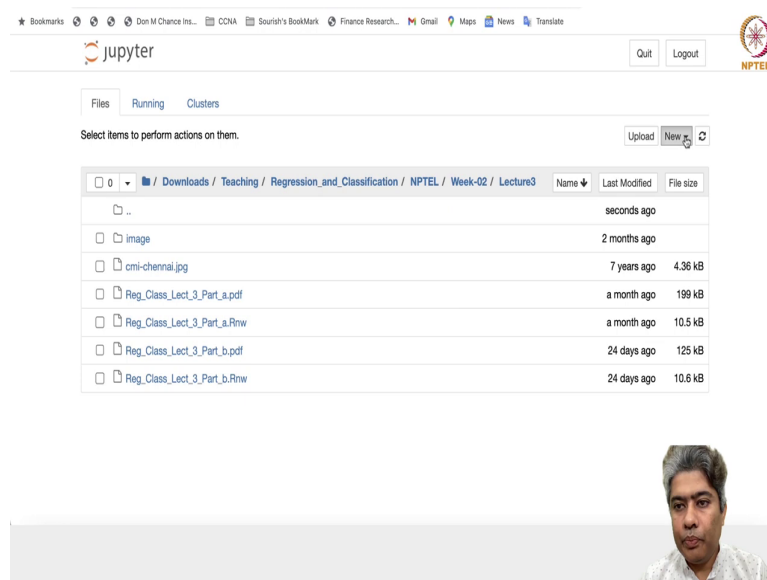


**Predictive Analytics - Regression and Classification**  
**Prof. Sourish Das**  
**Department of Mathematics**  
**Chennai Mathematical Institute**

**Lecture - 13**  
**Hands-on with Python Part – 2**

Hello, all. Welcome back to Lecture 3, Part C. In this part we are going to do some Hands-on on linear regression using Python.

(Refer Slide Time: 00:27)



The screenshot shows a Jupyter Notebook interface. At the top, there is a browser address bar with several tabs open, including 'Don M Chance Ins...', 'CCNA', 'Sourish's BookMark', 'Finance Research...', 'Gmail', 'Maps', 'News', and 'Translate'. The Jupyter logo is visible on the left, and 'Quit' and 'Logout' buttons are on the right. Below the navigation bar, there are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' with 'Upload' and 'New' buttons. The main area displays a file browser view for the path: 'Downloads / Teaching / Regression\_and\_Classification / NPTEL / Week-02 / Lecture3'. The file list is as follows:

	Name	Last Modified	File size
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	image	2 months ago	
<input type="checkbox"/>	cmi-chennai.jpg	7 years ago	4.36 kB
<input type="checkbox"/>	Reg_Class_Lect_3_Part_a.pdf	a month ago	199 kB
<input type="checkbox"/>	Reg_Class_Lect_3_Part_a.Rnw	a month ago	10.5 kB
<input type="checkbox"/>	Reg_Class_Lect_3_Part_b.pdf	24 days ago	125 kB
<input type="checkbox"/>	Reg_Class_Lect_3_Part_b.Rnw	24 days ago	10.6 kB

At the bottom right of the interface, there is a small video feed showing a man's face.

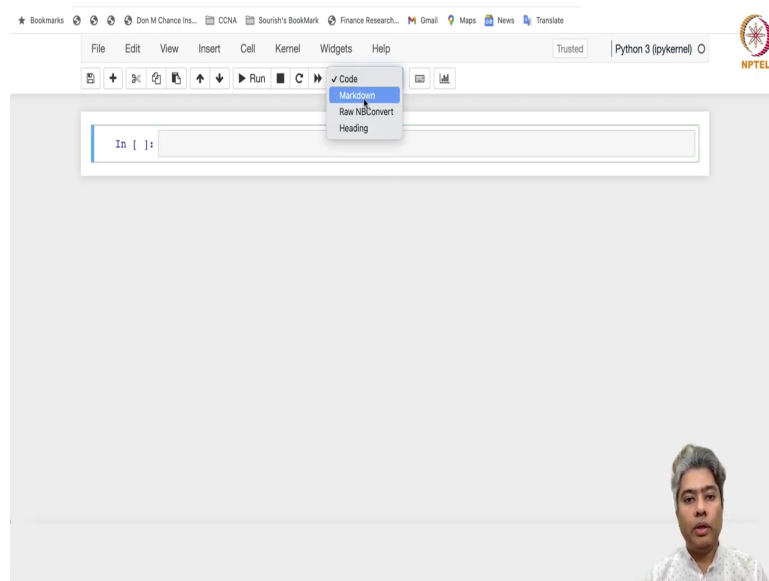
So, now first we will open our Jupyter Notebook.

(Refer Slide Time: 00:32)

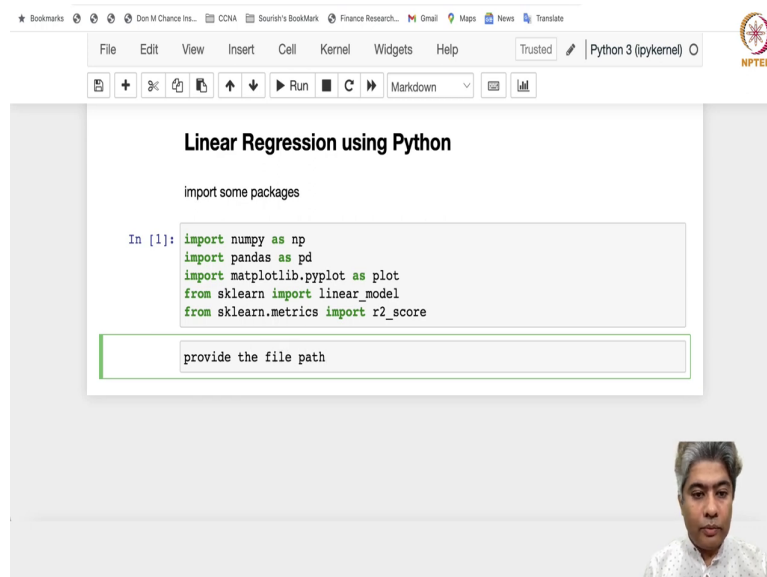
The screenshot displays the JupyterLab web interface. At the top, there is a browser address bar with several tabs and a search bar. Below the browser bar, the JupyterLab logo is visible on the left, and 'Quit' and 'Logout' buttons are on the right. The main area is divided into 'Files', 'Running', and 'Clusters' tabs. The 'Files' tab is active, showing a file browser for the path: `Downloads / Teaching / Regression_and_Classification / NPTEL / Week-02 / Lecture3`. A dropdown menu is open over the 'New' button, listing options: 'Notebook', 'Julia 1.6.3', 'Julia 1.8.2', 'Python 3 (pykernel)', and 'R'. Under the 'Other:' section, there are 'Text File', 'Folder', and 'Terminal' options. The file browser shows a list of files including 'image', 'cmi-chennai.jpg', and several PDF files related to 'Reg\_Class\_Lect\_3\_Part\_a' and 'Part\_b'. A video inset in the bottom right corner shows a man speaking, with a URL bar below it displaying `localhost:8888/tree/Downloads/Teaching/Regression_and_Classification/.../Lectu...`.

Here I am going to open Python 3 kernel.

(Refer Slide Time: 00:36)



(Refer Slide Time: 00:44)



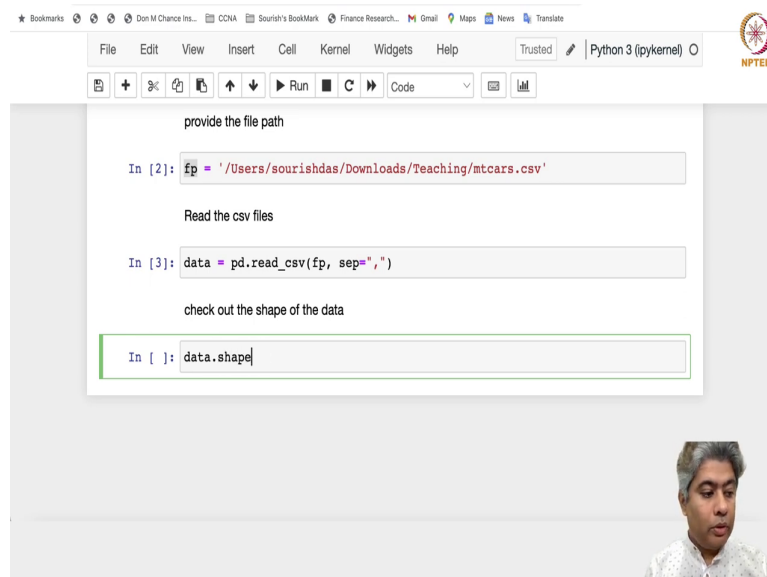
So, the first thing we are going to do is we have to first open some we will put some headings say Linear Regression using Python, ok and then I will write some more markdown. Say some import some packages import some packages. So, what are the packages we are going to import?

We are going to import numpy pandas for plot we are going to import some mac plot libraries and scikit learn or sk learn. So, from import we are going to import sorry import numpy as np, then import pandas as pd, then import mat plot library matplotlib dot particularly pyplot require we do not need to import everything as plot.

And, then from sk learn sk learn import linear model linear underscore model and from sklearn dot actually there is n missing sklearn dot metrics import r 2 score. So, r 2 score is going to calculate the r square. We have not studied the r square yet in the theory part we are going to discuss the model evaluation of the r square very soon in next few lectures.

But, we will figure out in this python hands on how to calculate r square from a fitted model. So, let us run this. So, all the packages are now imported. So, let me try to increase the size a bit I hope this size is ok with you guys. So, first provide the file path provide the file path.

(Refer Slide Time: 03:45)



The screenshot shows a Jupyter Notebook interface with the following content:

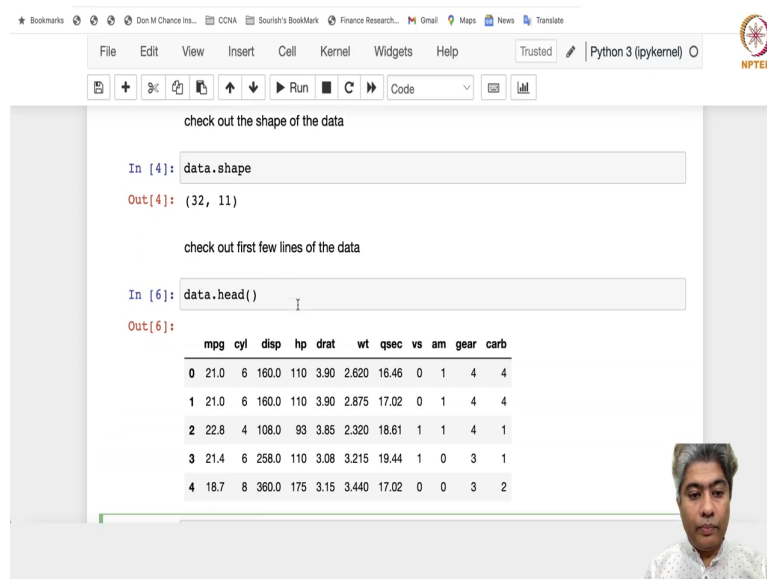
```
provide the file path  
In [2]: fp = '/Users/sourishdas/Downloads/Teaching/mtcars.csv'  
  
Read the csv files  
In [3]: data = pd.read_csv(fp, sep=",")  
  
check out the shape of the data  
In [ ]: data.shape
```

The interface includes a browser address bar at the top with various bookmarks, a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), and a toolbar with icons for file operations and execution. The Python 3 (ipykernel) logo is visible in the top right corner. An NPTEL logo is also present. A small video feed of a person is visible in the bottom right corner of the notebook area.

So, fp equals to factor of Users sourishdas. So, you have to in this path you have to give the path of the mt cars dot csv file. So, here the final file is mt cars dot csv and you have to give the full path so that it can be called and then read the csv files. So, here read the csv files and then data I am going to read the data pd dot read under square csv and I have to give the final path and separated by comma.

So, without any difficulty data is being read now we are going to check out the first shape of the data check out the shape of the data, ok.

(Refer Slide Time: 05:26)



The screenshot shows a Jupyter Notebook interface with a browser window at the top. The notebook contains two code cells. The first cell is titled "check out the shape of the data" and contains the code `data.shape`. The output is `(32, 11)`. The second cell is titled "check out first few lines of the data" and contains the code `data.head()`. The output is a table with 5 rows and 11 columns. The columns are labeled: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. The rows are indexed 0 to 4.

```
check out the shape of the data

In [4]: data.shape
Out[4]: (32, 11)

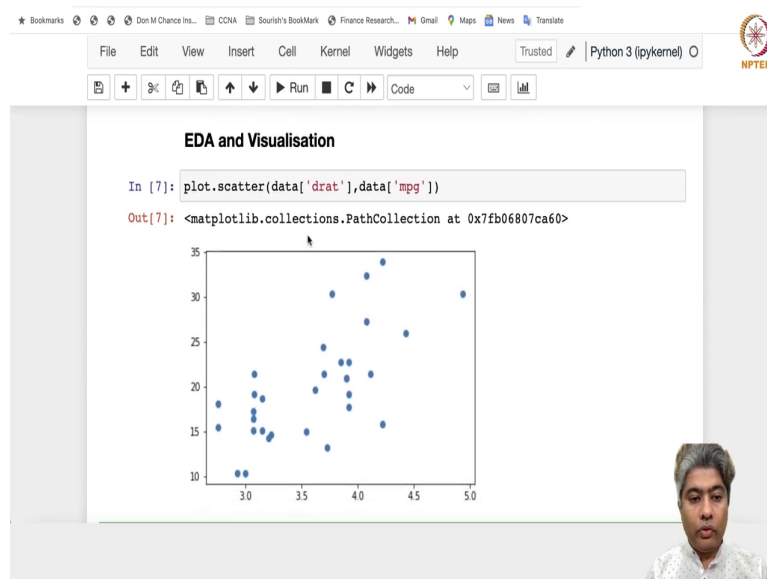
check out first few lines of the data

In [6]: data.head()
Out[6]:
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

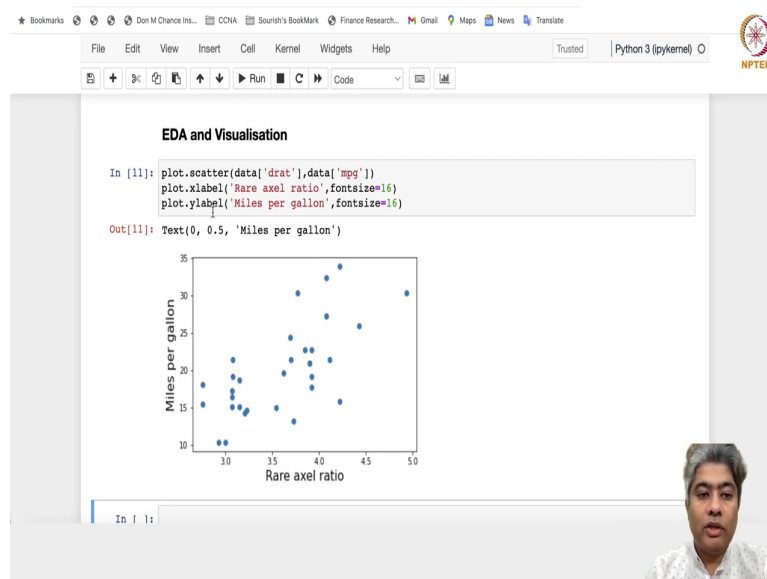
So, data dot shape is 32 cross 11. So, we have in this data we have 32 rows and 11 columns. So, check out first few lines check out first few lines of the data ok. Sorry, it has to be a markdown yeah now data dot head is the one. So, theta data dot head gives you the first five lines of the data frame that you are handling. So, here miles per gallon is our going to be our target variable and cylinder displacement, horsepower, rare axel ratio these are going to be the our independent variables.

(Refer Slide Time: 06:27)



Now, first what we have to do? We have to do some EDA and visualisation EDA and visualisation by. So, I will put some mark on that maybe a third. Now, so, first we make some scatter plot. So, data drat and data mpg if I run this so, you can see this it is plotting the data, but it has not given any x-axis or y-axis. So, we have to give a name to the label we have to provide a label to x-axis and y-axis.

(Refer Slide Time: 07:20)



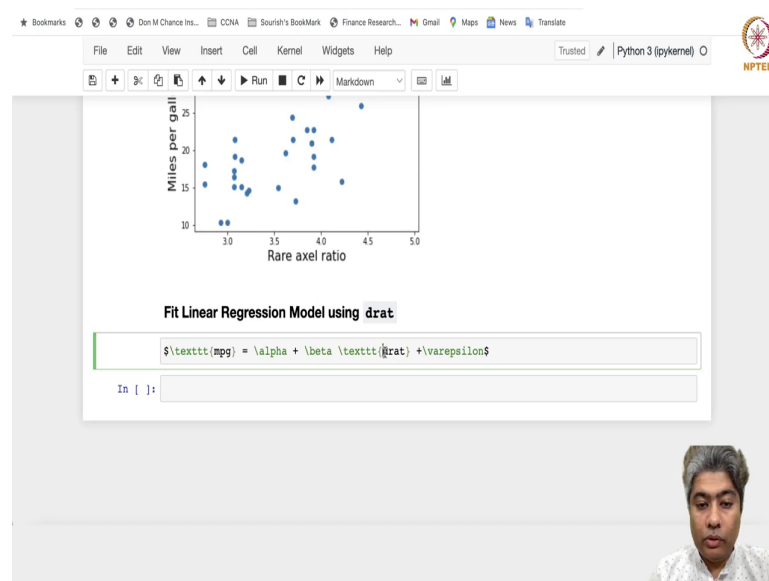
So, plot dot xlab we have to say what is the what is the in the x axis this is fair absolute ratio we have to give a font size maybe 16 and similarly, we have to give Miles per gallon. So, now if I Run it, sorry, we have to be xlabel there has to be misspelling error ylabel. Yeah, now it is fine. So, let me just a little bit smaller.

So, on the y-axis we have miles per gallon and on the x-axis we have the rare axel ratio, ok. So, if you do not put this semicolon you can see this text 0, 0.5 Miles per gallon is coming. The last you know some message about the last line, but if you put this semicolon here, then you will see it will not appear and it will look little better.

Next we are going to fit the linear regression model with rare axel ratio.



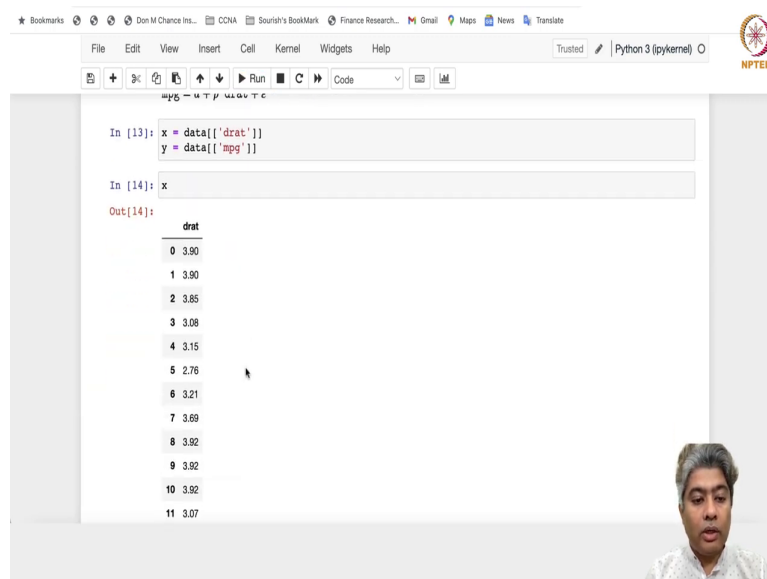
(Refer Slide Time: 09:02)



So, right Fit Linear Regression Model using drat ratio. So, maybe you put it like this you can see it looks like the variable itself and some theory you can put about first we have to put your markdown and then we can say that ok miles per gallon miles per gallon is a function of alpha plus beta times t plus varepsilon.

So, if you Run it, so, you can see miles per gallon as a function of alpha plus beta times rare axel ratio. So, you can put a little bit of space so that you have the things much more clearly nicely done.

(Refer Slide Time: 10:20)



```
In [13]: x = data['drat']
y = data['mpg']

In [14]: x

Out[14]:
```

	drat
0	3.90
1	3.90
2	3.85
3	3.08
4	3.15
5	2.76
6	3.21
7	3.69
8	3.92
9	3.92
10	3.92
11	3.07

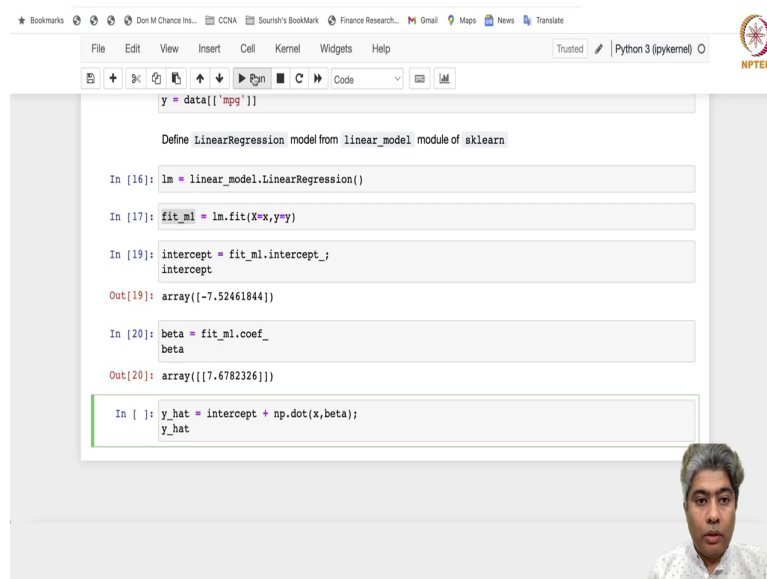
Our job is to calculate the value of alpha and beta from the data. So, first thing is what is our x? Our x is drat the ratio. So, we will call it as a array we will define it as array and y is miles per gallon.

(Refer Slide Time: 10:54)

```
3 3.08
4 3.15
5 2.76
6 3.21
7 3.69
8 3.92
9 3.92
10 3.92
11 3.07
12 3.07
13 3.07
14 2.93
15 3.00
16 3.23
17 4.08
18 4.93
19 4.22
20 3.70
```

Now, if I just say Run x you can see this is as array or y you can see this is a single array. It is being read as a single array.

(Refer Slide Time: 11:08)



```
★ Bookmarks  Don M Chance Ins...  CCNA  Sourish's BookMark  Finance Researc...  Gmail  Maps  News  Translate

File  Edit  View  Insert  Cell  Kernel  Widgets  Help  Trusted  Python 3 (pykernel) 0  NPTEL

y = data[["mpg"]]

Define LinearRegression model from linear_model module of sklearn

In [16]: lm = linear_model.LinearRegression()

In [17]: fit_m1 = lm.fit(X=x,y=y)

In [19]: intercept = fit_m1.intercept_
intercept
Out[19]: array([-7.52461844])

In [20]: beta = fit_m1.coef_
beta
Out[20]: array([[7.6782326]])

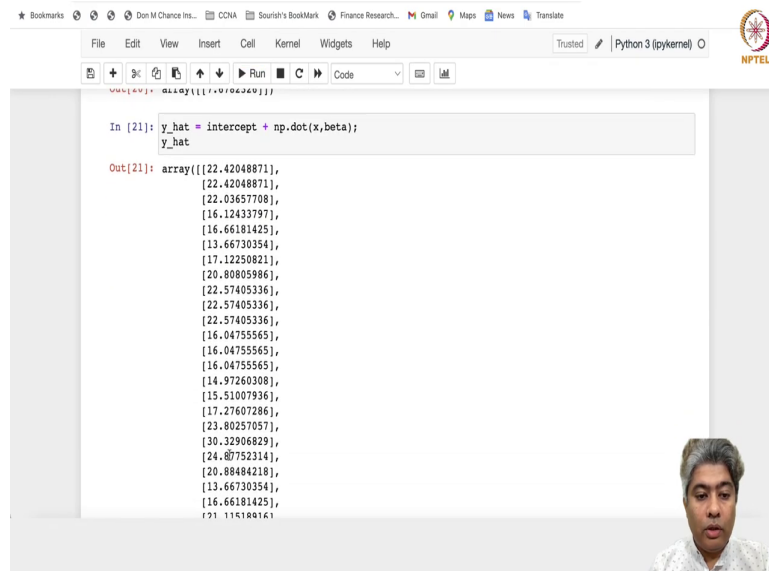
In [ ]: y_hat = intercept + np.dot(x,beta);
y_hat
```

Now, next first we have to define the linear regression from module from the scikit learn define probably we just write define linear regression model from linear model module linear model module of sklearn package. So, this is our first task and how we define this In equals to linear model. So, remember that in the beginning we have from sklearn we have imported linear underscore model. From linear underscore model we are defining linear recreation, ok and we defining it as a ln, ok.

And, now we are going to fit our first fit our model fit m1 joint of lm dot fit X equal to x and y equals to y. So, it fitted the value and now from the model if we just say intercept underscore. So, this is our intercept. This value is our intercept and similarly, if we just say beta equal to fit underscore m1 dot coef underscore it will give you the coefficient value of beta 7.6782.

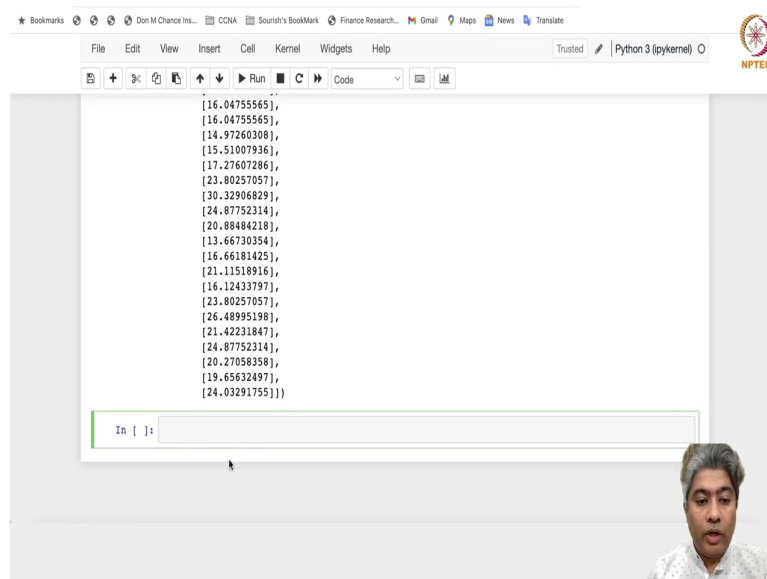
Now, we are going to calculate the  $\hat{y}$  values  $\hat{y}$  values are intercept plus np dot product between  $x$  and  $\beta$ . So, if we just can write.

(Refer Slide Time: 14:20)



```
Out[21]: array([[22.42048871],
 [22.42048871],
 [22.03657708],
 [16.12433797],
 [16.66181425],
 [13.66730354],
 [17.12250821],
 [20.80805986],
 [22.57405336],
 [22.57405336],
 [16.04755565],
 [16.04755565],
 [16.04755565],
 [14.97260308],
 [15.51007936],
 [17.27607286],
 [23.80257057],
 [30.32906829],
 [24.8752314],
 [20.88484218],
 [13.66730354],
 [16.66181425],
 [21.11518816]])
```

(Refer Slide Time: 14:22)

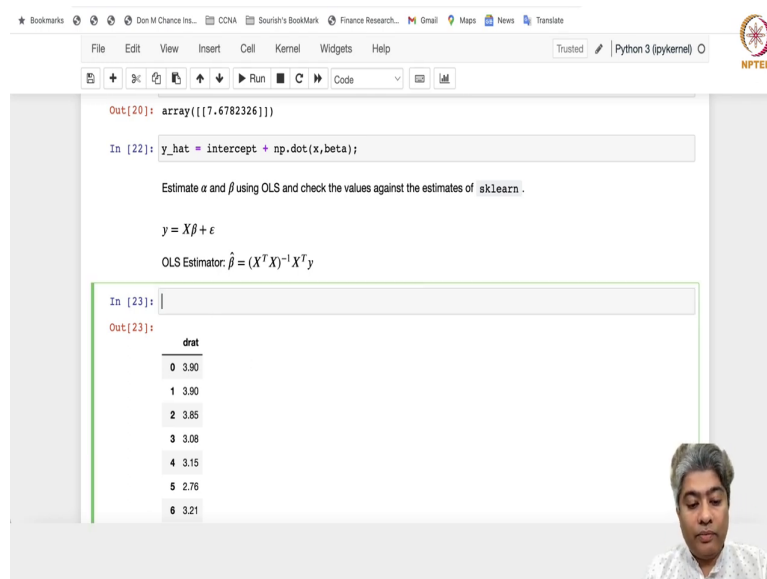


```
[16.04755565],  
[16.04755565],  
[14.97260308],  
[15.51007936],  
[17.27607286],  
[23.80257057],  
[30.32906829],  
[24.87752314],  
[20.88484218],  
[13.66730354],  
[16.66181425],  
[21.11518916],  
[16.12433797],  
[23.80257057],  
[26.48995198],  
[21.42231847],  
[24.87752314],  
[20.27058358],  
[19.65632497],  
[24.03291755]]
```

In [ ]:

So, these are fitted values ok these are fitted values. Now, what we want we want to check that we also learned that this model can be fitted using OLS method and in the OLS method using OLS method also we can calculate how the value of alpha and beta. And, we will use OLS method to calculate the value of alpha and beta and we will check if the value of our estimated method using OLS is matching with the sklearn's estimated values.

(Refer Slide Time: 15:12)



The screenshot shows a Jupyter Notebook interface with the following content:

```
Out[20]: array([[7.6782326]])
```

```
In [22]: y_hat = intercept + np.dot(x,beta);
```

Estimate  $\alpha$  and  $\beta$  using OLS and check the values against the estimates of sklearn .

$$y = X\beta + \epsilon$$

OLS Estimator:  $\hat{\beta} = (X^T X)^{-1} X^T y$

```
In [23]:
```

```
Out[23]:
```

	drat
0	3.90
1	3.90
2	3.85
3	3.08
4	3.15
5	2.76
6	3.21

A small video feed of a person is visible in the bottom right corner of the notebook interface.

So, estimate alpha beta alpha and beta using OLS and check the values against the sklearn estimates of estimates of sklearn. Technically they should match, but let us try to check it out. So, what is our OLS method? OLS method is  $y$  equal to  $X$  beta plus varepsilon and OLS estimator is  $\hat{\beta}$  equals to  $X$  transpose  $X$  inverse  $X$  transpose  $y$ .

So, we are going to use exactly this formula to estimate the values of alpha and beta. Our estimated beta hat will be a 2-dimensional vector with the first component of the vector would be the estimate of alpha or intercept and the second component would be the estimate of slope.

So, let us try to estimate the alphas and betas. So, first let us see how our  $x$  looks like. So, our  $x$  is a simple one dim 32 dimension vector or in a metrics terms is a 32 cross 1 column.

(Refer Slide Time: 17:37)

The screenshot shows a Jupyter Notebook interface with a browser window at the top. The browser's address bar shows several tabs, including "Don M Chance Ins...", "CCNA", "Sourish's BookMark", "Finance Research...", "Gmail", "Maps", "News", and "Translate". The notebook's menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The current kernel is "Python 3 (pykernel)".

The notebook content includes the following text and code:

OLS Estimator:  $\hat{\beta} = (X^T X)^{-1} X^T y$

First define the design matrix  $X$

```
In [26]: n = data.shape[0]
n
Out[26]: 32
```

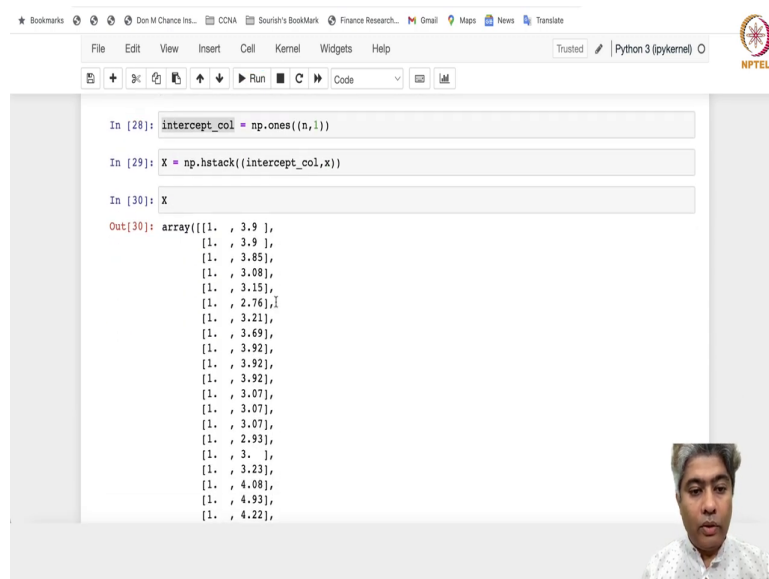
```
In [27]: intercept_col = np.ones((n,1))
intercept_col
Out[27]: array([[1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.],
 [1.]])
```

In the bottom right corner of the notebook, there is a small video feed window showing a person's face.





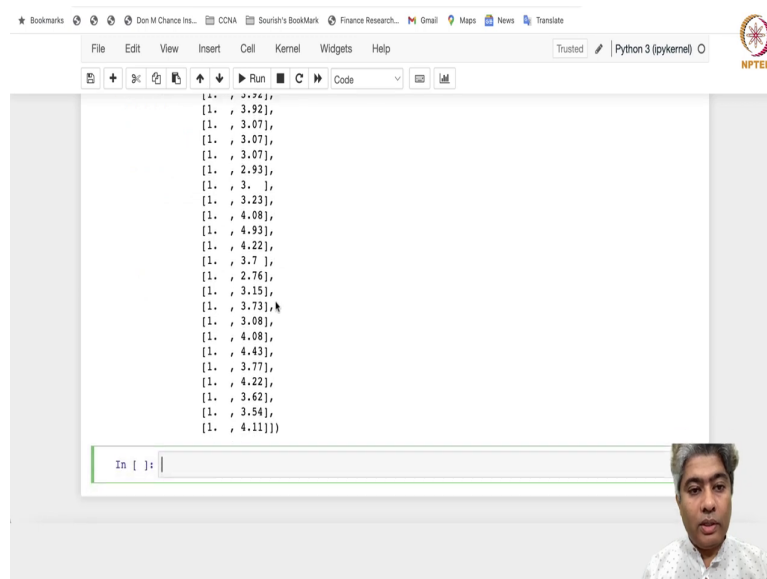
(Refer Slide Time: 19:06)



```
In [28]: intercept_col = np.ones((n,1))
In [29]: X = np.hstack((intercept_col,x))
In [30]: X
Out[30]: array([[1. , 3.9 ],
                [1. , 3.9 ],
                [1. , 3.85],
                [1. , 3.08],
                [1. , 3.15],
                [1. , 2.76],
                [1. , 3.21],
                [1. , 3.69],
                [1. , 3.92],
                [1. , 3.92],
                [1. , 3.92],
                [1. , 3.07],
                [1. , 3.07],
                [1. , 3.07],
                [1. , 2.93],
                [1. , 3. ],
                [1. , 3.23],
                [1. , 4.08],
                [1. , 4.93],
                [1. , 4.22],
```



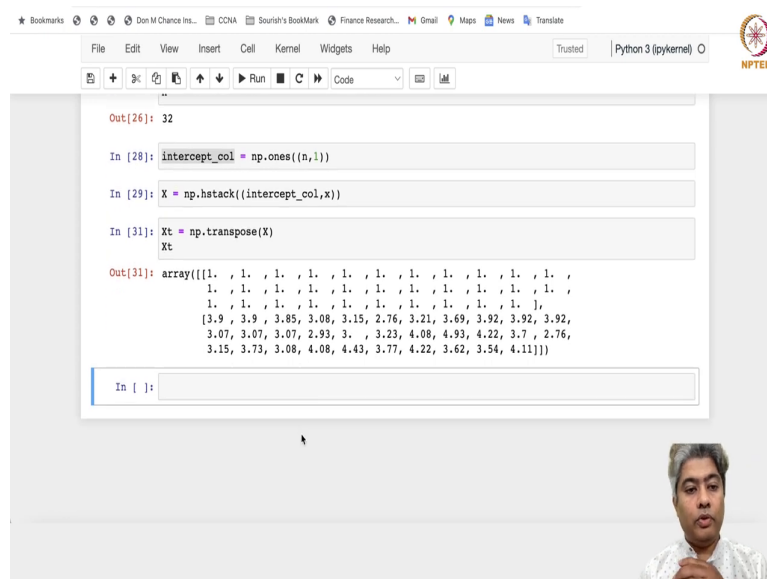
(Refer Slide Time: 19:28)



```
Out[ ]:
[[1. , 3.92],
 [1. , 3.07],
 [1. , 3.07],
 [1. , 3.07],
 [1. , 2.93],
 [1. , 3.  ],
 [1. , 3.23],
 [1. , 4.08],
 [1. , 4.93],
 [1. , 4.22],
 [1. , 3.7  ],
 [1. , 2.76],
 [1. , 3.15],
 [1. , 3.73],
 [1. , 3.08],
 [1. , 4.08],
 [1. , 4.43],
 [1. , 3.77],
 [1. , 4.22],
 [1. , 3.62],
 [1. , 3.54],
 [1. , 4.11]]
```

And, capital X is equal to np dot hstack sorry, first is intercept column comma, x. Now, if you look into it you can see it is a 32 rows comma 2 column. So, first column is all intercept and the second column is all rare axel ratio, alright.

(Refer Slide Time: 19:43)



The screenshot displays a Jupyter Notebook interface with the following content:

```
Out[26]: 32
```


```
In [28]: intercept_col = np.ones((n,1))
```

```
In [29]: X = np.hstack((intercept_col,x))
```

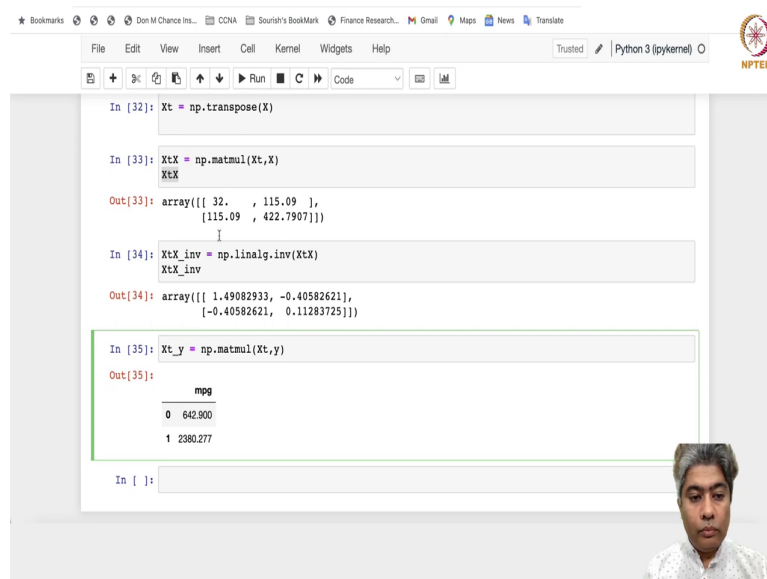
```
In [31]: Xt = np.transpose(X)
Xt
```

```
Out[31]: array([[1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. ,
1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. ,
1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. ],
[3.9 , 3.9 , 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.92, 3.92,
3.07, 3.07, 3.07, 2.93, 3. , 3.23, 4.08, 4.93, 4.22, 3.7 , 2.76,
3.15, 3.73, 3.08, 4.08, 4.43, 3.77, 4.22, 3.62, 3.54, 4.11]])
```

In [ ]:



(Refer Slide Time: 20:03)



```
In [32]: Xt = np.transpose(X)

In [33]: XtX = np.matmul(Xt,X)
XtX
Out[33]: array([[ 32.    , 115.09 ],
               [115.09 , 422.7907]])

In [34]: XtX_inv = np.linalg.inv(XtX)
XtX_inv
Out[34]: array([[ 1.49082933, -0.40582621],
               [-0.40582621,  0.11283725]])

In [35]: Xt_y = np.matmul(Xt,y)
Out[35]:
      mpg
0    642.900
1    2380.277

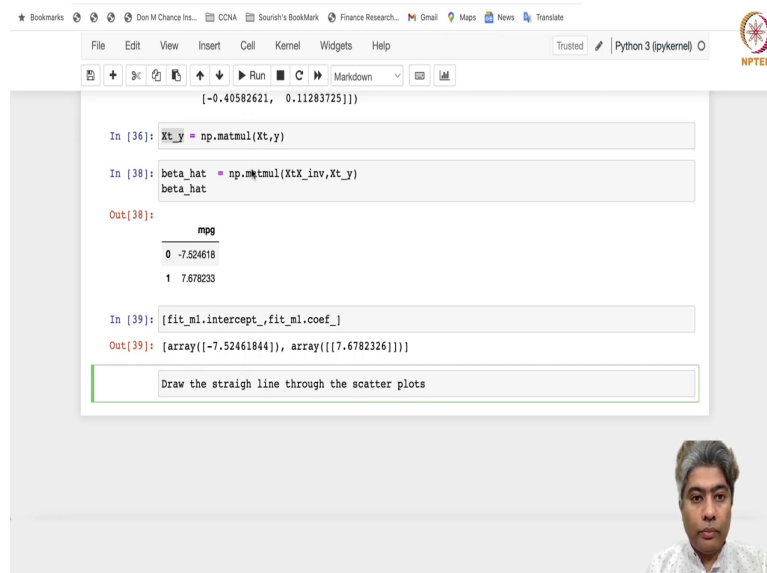
In [ ]:
```

Now, we have to compute the X transpose  $X^T$ . So, this is our X transpose matrix and then we have to calculate the X transpose X matrix. So, basically this is a multiplication between X transpose and X. So, it will be np dot matrix multiplication from numpy we will call this matmul function and you will just provide X 2 function that we want to calc. No multiply and this is my final X transpose X.

Now, what we have to do? We have to calculate the X transpose X inverse is for that from numpy is linear algebra module. We have to call the inverse and we have to provide X transpose X and so, that is this is our X transpose X inverse. Now, if we look at it is X transpose X inverse X transpose y, now X transpose is a 2 cross n and y is a n cross 1. So, X transpose y will be 2 cross 1 product.

So, let us calculate the X transpose y. So, np dot matrix multiplication X transpose and y if I do that so, this is my X transpose y. So, I can just calculate that.

(Refer Slide Time: 21:51)



```
[-0.40582621, 0.11283725]]

In [36]: Xt_y = np.matmul(Xt,y)

In [38]: beta_hat = np.matmul(XtX_inv,Xt_y)
beta_hat

Out[38]:
      mpg
0 -7.524618
1  7.678233

In [39]: [fit_m1.intercept_,fit_m1.coef_]

Out[39]: [array([-7.52461844]), array([[7.6782326]])]

Draw the straight line through the scatter plots
```

And, now what I have to do? I have to just do another matrix multiplication np dot matrix multiplication within X transpose X inverse and X transpose y. If I do that, so, this gives me the beta hat. So, beta hat is this. Now, this is the OLS estimate we need to check against the value of area of interest from the sk learn. So, from sk learn we just call this and fit m1 dot coef. So, let us Run this.

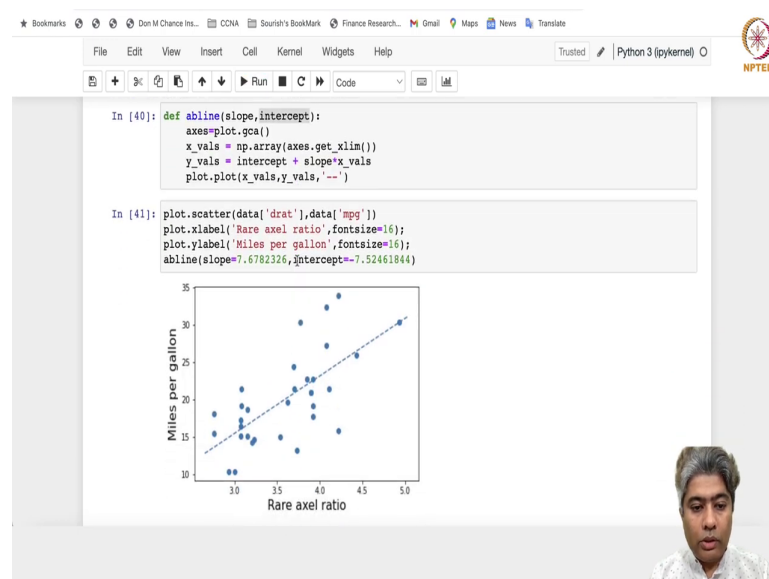
So, these values are the first value is the intercept value negative second 7.52467. So, it is exactly matching the OLS method with the sk learn. So, sk learn itself is also is implementing

OLS method ordinary least square method and 7 point slope is 7.678. So, we see how you can calculate the analytic solution using OLS method for simple linear regression.

Now, what we will do we will, now since we have the alphas and betas we will draw a straight line through these scatter plots. For that what we have to do? We have to calculate a we have to write a ab line function. So, in r there is a nice function called ab line we will write a small function called ab line and we will call that and we will use that to draw the straight line through these scatter plots.

So, we will draw the straight line through a scatter plots ok.

(Refer Slide Time: 24:26)

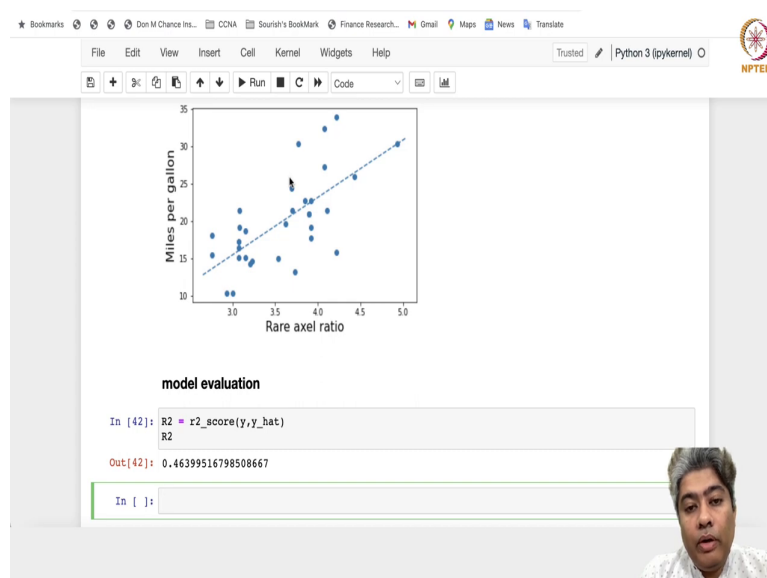


Now, for that abline slope comma intercept the first we have to define the axes. So, we call the axes from plot dot gca and then we will check out the x values that np dot array from the

axes. We will get the x limbs or x values and y vals will be simply. So, this could be the x vals, the y values will be simply intercept plus slope times x values and then plot dot plot x vals comma y vals and we want dot dot y.

So, let us the np line is ready now. What we will do? We will just pick this pieces flying here and without disturbing it you will just go and plot it there and then abline is equal to slope is. So, this is the value of slope equal to this comma intercept equal to. So, this is the value of intercept. So, let me just Run this. Now, you can see this is the OLS fitted line through the scatter plots.

(Refer Slide Time: 27:05)



Finally, we will do a little bit of model evaluation. I will just introduce the concept but using how can you do this using a model evaluation because model evaluation is extremely important component whenever you are fitting a linear regression you should do a model



evaluation. So, let us and  $r^2$  is one of the metric very popular metric and we have imported  $r^2$  score from the matrix `sk learn matrix` module.

So, we will what we will do, we will just call that  $R^2$  equal to  $r^2$  score  $y$  comma  $\hat{y}$  and then just check it out. So, it is  $R^2$  for this fitted model is 46.39 or about 46.4 percent. So, that means, about if you look into the miles per gallon whatever variability you have about 46 percent of the variability can be explained by the rare axle ratio which is a very good thing.

And, so, we will talk more about  $R^2$  in the coming lectures and you can do lot of things. We will with the you know with the `sk learning python`. So, we will stop here today and next well see you at Lecture 4.

Thank you very much. Bye.