Business Analytics & Data Mining Modeling Using R

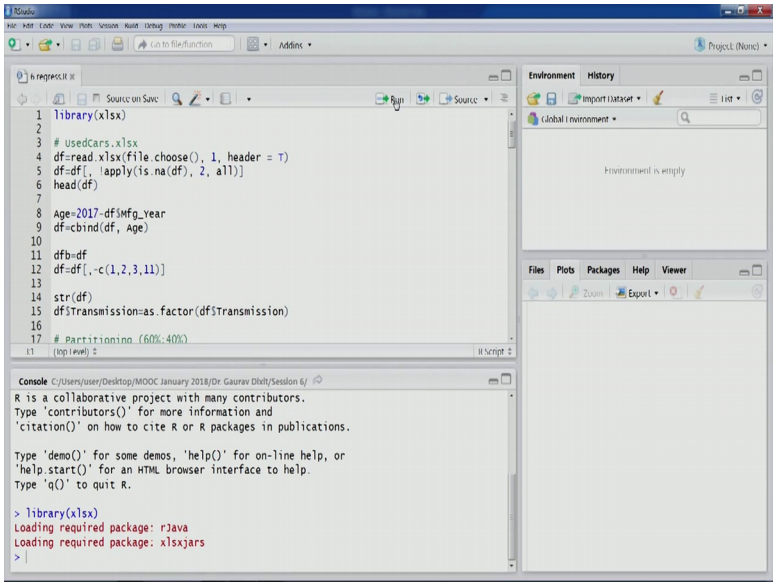
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture – 23
Multiple Linear Regression-Parts II

Welcome to the course business analytics and data mining modeling using R. So, in the previous lecture we started our discussion on this particular technique multiple linear regression. So, we discussed the linear regression the equation, the different coefficient that we are required to estimates. We talked about the exploratory modeling and the predictive modeling a few differences and we also try to understand the application of this particular technique multiple linear regression, in a statistical technique and how it is different in a data mining environment. So, we also talked about some of the assumptions you know that when we apply OLS to estimate those coefficient those beta zeros and sigma.

Then what are the underlying assumption that we have to follow, and how those assumptions are different you know how in first assumption especially noise following normal distribution, how we get some relaxation relaxation for from that assumption in a data mining setting. We talked about all those things then we will again go through an exercise to understand how linear regression modeling is done and how different concept can be put into practice.

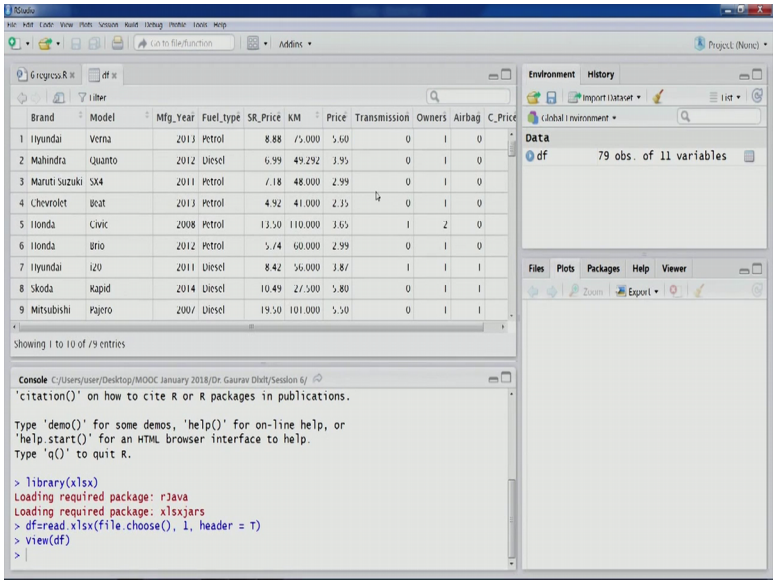
(Refer Slide Time: 01:54)



So, let us open R studio. So, as usual we will load this particular library x l s x because the data that the this particular data used cars data set, that we have it is in the excel file.

So, let us import this data set. So, the function that we are going to use is read dot x l sx first argument is as usual we are going to browse for this particular file and then the first worksheet will import the data of the first worksheet and the header is true because we have name of all the variables there in the data set. So, let us execute this line you can see in the environment section this particular data set has been imported and you can see 79 observation of eleven variables.

(Refer Slide Time: 02:54)

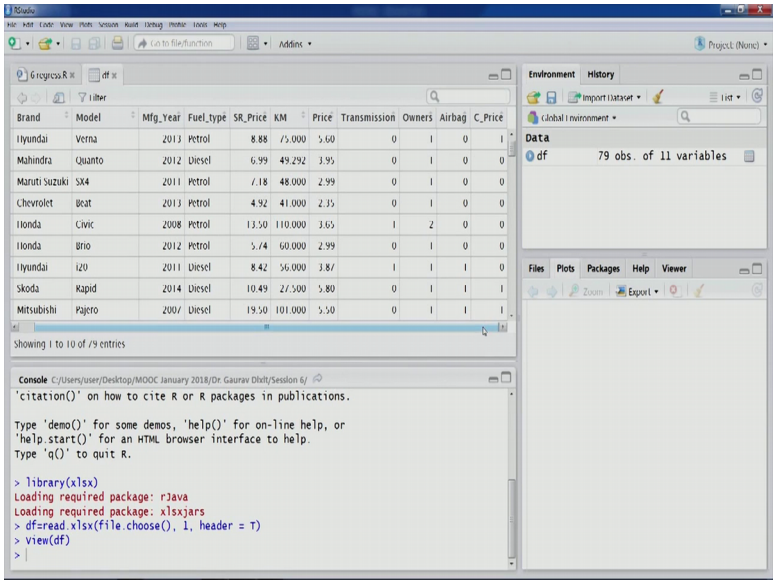


Let us have a look at this data in the R environment. So, this is small icon that you see in the environment section once the data has been imported. So, once you click on this particular icon you would be able to see another tab that would open and you would be able to see the data just like you see it in the excel files. So, you can see the variable names brand models, manufacturing, your fuel types. So, there are 3 fuel types for these used cars petrol diesel CNG and then we have showroom price for each of these used cars. So, when these cars were what first time what; these right as the new cars. So, what was the price and then the kilometers since these cars have been running on road, the kilometers that have been accumulated from the starting years from the purchase year

And then the price this particular price is the used the offered price for the these used cars. The cars whether the transmission is manual or automatic that is also we have information on. So, 0 representing the manual, and one representing the automatic transmission.

Now, next variable is on owners where each number is representing the number of owners that actually owned the number of people number of individuals who actually owned this particular car. So, that is also there. You also have information on some of the security features for example airbags.

(Refer Slide Time: 04:31)



So, number of airbags that are there in the car. So, that that information is also available.

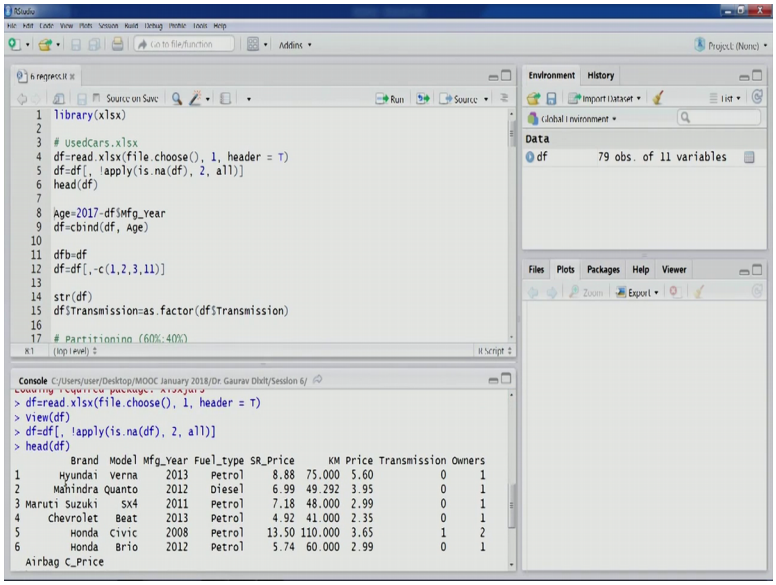
We have another variable in the data set that is c prices, that is this is this variable what is generally present for the classification task where any car having a less than offered value of a 4 lakhs is represented by 0 and the cars having value of equal to or more than 4 lakhs are represented by 1 So, this is the data set. So, let us close this particular tab and let us. So, first thing would be as usual we would like to remove the n a columns.

So, we used these within the brackets and the for the column value. So, we have applied this first particular function. So, apply what is going it is going to do is it will find out the using is dot n a function will find out which of the columns in this particular data frame d f d f they are having na values right. So, those particular columns would be selected. So, 2 is indicating that this particular function is being applied or you know column wise on this particular data frame.

So, therefore, those particular columns na columns would be selected and the all function would then be applied on those columns. So, so other columns which have which do not have na values, they would be true and then the columns which I have na values which will have false values. A logical operator logical vector would be written from this function and then we have this another operator not that would apply on this logical vector logical result, and then all the true values would be converted to false and all the false values would be converted into true. So, therefore, all the n a columns which were identified using the apply function and they were indicated as false now they would become.

Now,. So, the reverse will happen. So, the all the columns which do not have na values they would be returned as false using apply function, and when not is applied then they would become true, and the other columns they would be returned as true because they were na columns and when not is applied they will become false to those columns would actually be dropped. So, this is how this particular line we have been using quite often. So, this is how it will operate. So, in this particular data set we did not have any such column. So, the result would remain same.

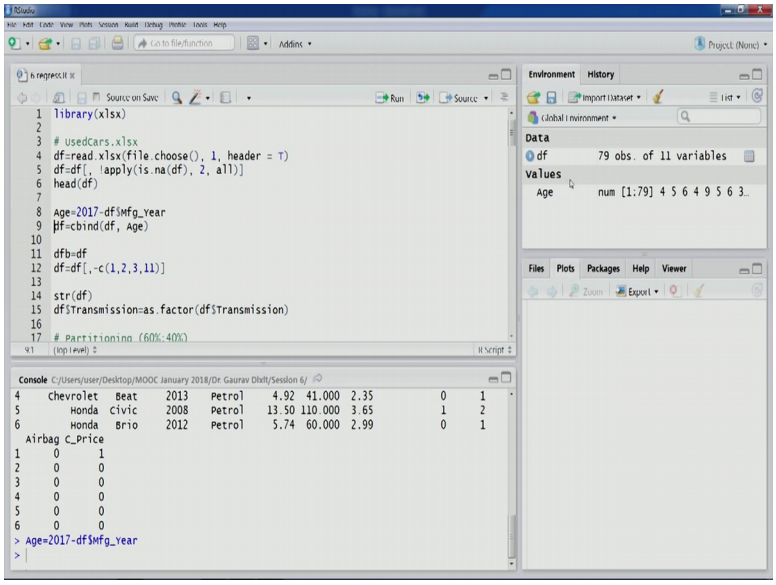
(Refer Slide Time: 07:40)



Now, let us look at the first 6 observation of this particular dataset, you can see the name these variables as we saw through other options in R studio by clicking this particular icon, and looking at the full file in one go. So, using head function we can look at the 6 observation we need not look at all the observation, that would require the whole file to be loaded into memory and therefore, if your device where you are running your r studio and if it does not have sufficient ram built into it then probably you would are you are better of running head function, and you would just be loading 6 observations into memory.

So, these are the observations the variables we have already discussed. So, one particular third column that we can see is manufacturing here, when the car was actually manufactured. So, the age of car can actually be computed using this particular vector. So, this is what we are going to do next. So, you can see age variable and if the if we if these offered prices if all the information is in the context of year 2017. So, therefore, the current here is 2017. So, we can subtract all this particular vector from 2017 and all the values all the observations for all the observations will get the difference and therefore, the age.

(Refer Slide Time: 09:19)



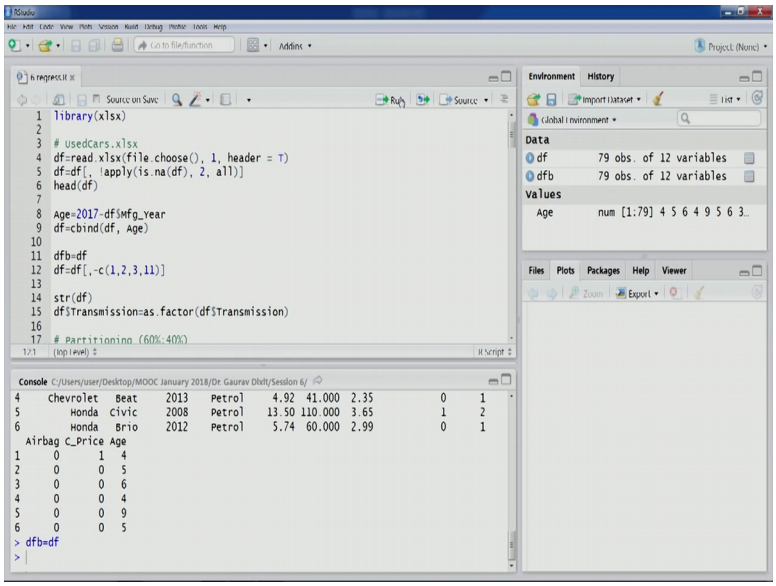
So, let us execute this line you can see in the environment section age variable has been created this numeric vector, having 79 values right. So, age of all the used cars have been computed. Now age could be a relevant variable because as we discussed the task here is the prediction task we are trying to we would be trying to build a model, trying to we will build a model to predict the price of a used car right offered the price of a used car. So, age could be an important variable in terms of explaining in terms of predicting that particular price. So, therefore, you would like to have age also in our model.

So, let us append this particular variable this particular vector and this data frame. So, c bind is the function as we have talked about in previous lectures also, this can be used to an already existing you know a data frame to append this particular variable. By default it would append this variable at the end of all the all other columns in the data frame. So, let us execute this line. So, now, this particular variable has been appended. If you want to check again you can run the head you can call the head function again and you would see the last column age has been created and you can also see the values for first 6 observations.

Now, if we look at this particular data set, then brand name first 2 columns brand name model and also third one also manufacturing here they do not seem to be relevant for our analysis for our manufacturing year we have already we transformed this particular variable into an age variable, age of the used car. So, therefore, we would not be requiring this particular variable right. So, we will get rid of this variable also since we would be building a prediction model. So, therefore, this c_price which was the outcome variable mainly designed for the classification task, we also would not be requiring this variable.

So, we can get rid of these four variables; first one brand then model than manufacturing year and then this one c price, and the remaining variables would be the outcome variable that is price or the relevant predictors, that we want to include in our model. So, first let us take a backup of the existing data frame.

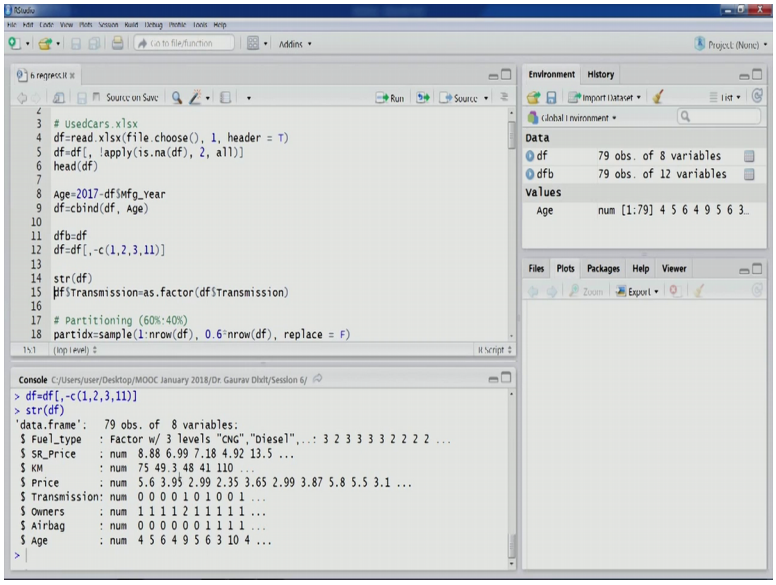
(Refer Slide Time: 12:01)



So let us take a backup you can see this particular data frame has been created now let us eliminate these columns. So, combined function can be used and minus before combined function indicating that we do not want to subset, that we want is other columns that are mentioned here in the combined function.

So, if you want to have a look at now if we are interested, we can have a look at the structure of the data frame, now in the present data frame this data frame that we have is we have all the variables of interest.

(Refer Slide Time: 12:44)



You can see now we have 79 observation of 8 variables. So, all the variables are now of interest to us whatever modeling exercise. So, fuel type right that would also help us determine the price of offered price of a used car because the CNG whether the car is running on a CNG fuel or diesel or petrol.

So, these cars are when they are purchased for the first time, they are priced differently and therefore, depreciation and other factors they work differently. Therefore, prices of these cars based on the fuel type that they have are going to be different therefore, fuel type being an important predictor for us in this prediction modeling exercise. Now as s r price is actually the show room price. So, this is equivalent of similar to the first time this particular and that particular car was purchased what was the price that was paid right this is the price over the years, the depreciation would be applied and the depending on the condition of the car and other variables some of them some of those variables, we have in this data set as well the offered price would be adjusted.

Now, how this offered price is being determined by the individual is part of this exercise of this model. Now the another important variable that we have is k m. So, the number of kilometers that a car has accumulated that also tells about the wear and tear that the car might have gone through, because of the those number of kilometers covered right. So, therefore, kilometer might also indicate the value of a car that is you know that should be depreciated that should become part of the depreciation. If the car has been driven less then probably it has not gone through that much of wear and tear.

But if it has traveled more than probably you know more wear and tear might have happened and therefore, the price might be on the lower side therefore, kilometer is also an important variable in this exercise. Price is the variable the offered price that is the variable that we are trying to predict this is the outcome variable of interest to us, next variable that we have is transmission. So, transmission right now as indicated in this output this is right now numerical vector as shown here, but this variable can have only 2 values or let us say 2 labels, because this is a categorical variable because we have just 2 labels whether the car is automatic or the car is manual.

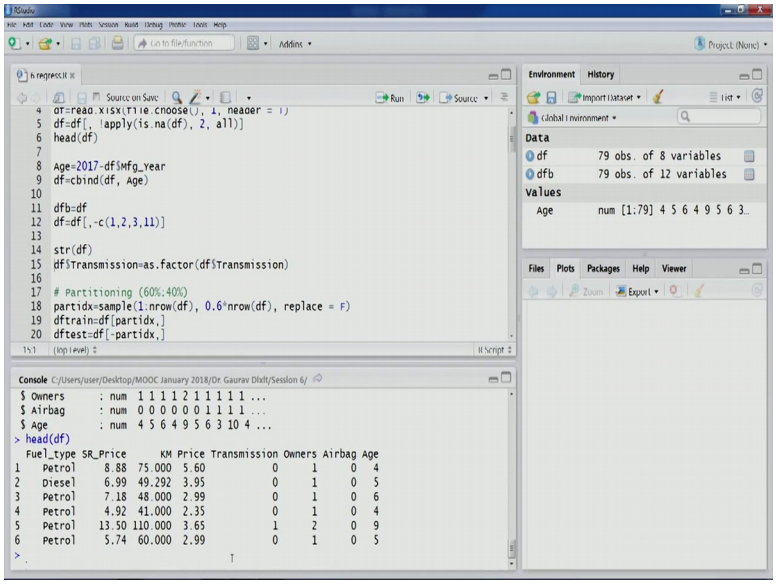
But here in this data set it is being shown in it is a numeric vector here zeros and ones. So, we would like to convert this vector from numeric to categorical variable to factor variable. So, therefore, you can see in the next line we have used as dot factor function to coerce this particular numeric variable into a factor variable in r environment. Because this variable has 2 labels 2 categories that is the car is automatic or manual.

So, let us execute this particular code and once it is done you can again run this structure function and you would see a change there the transmission variable, now you can see it has been converted into a factor variable with 2 levels 0 and 1.

So, now the values 0 and 1they are being treated appropriately because this being a categorical variable. So, now, the these values have become numeric codes indicating 2 different labels now in the structure function you would also see that the some few initial observation that are being shown here they are in this format one category 1, category 2 1 and 1 and then 2 and then 1, but the actual values at we had saw before in the using the head function they are either 0 and 1.

So, this is just the way output of a structure function is presented the values have not changed because of that conversion that we just did if you want if you are interested.

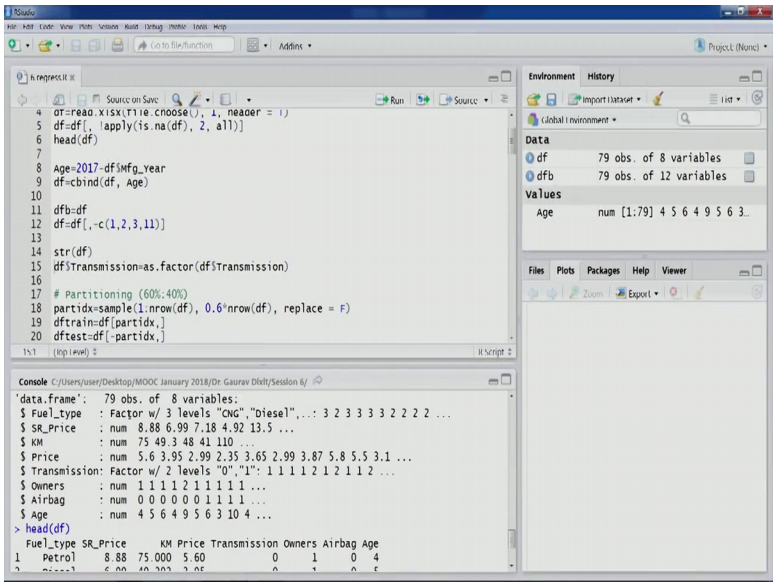
(Refer Slide Time: 17:36)



You can again have a look at the actual values you can see the transmission variable is still having the same 0 and values 0 and 1 value it is only the structural function that is presenting the output in that fashion, that for factor variables if for fact for factor variables it generally shows 1 2 and 3 for different categories, different labels that we might have. You can see even for the fuel type variable CNG diesel and petrol these are the 3 labels that we had the in the structure output we see 3 2 3

So, these representing the 3 classes 1 2 and 3, but actual values are same they are not disturbed. So, you can see fuel type you can see it the these strings these characters patrol diesel and therefore, those values are not changed it is just the representation in the output of the structure function.

(Refer Slide Time: 18:40)



Now, another important thing that we need to understand here is, that the factor variables that we see the are nominal variables the way they have been created these are nominal variables, they are not ordinal and the way labeling is done here.

For example if we check the be run d f of function for the first variable that is fuel type, you would see these values first 6 values petrol diesel, petrol petrol and petrol all and then labels CNG diesel and petrol these labeling in r environment is done alphabetically is therefore, CNG because it start with c which is and then diesel it is start with d and petrol p. So, the alphabetical in alphabetical fashion the ordering of labels is in that fashion, but when we do when we run a classification task, we will have to decide our reference category. So, in that case if we happen to select our reference category as petrol that would not be by default made as the reference category here in this case.

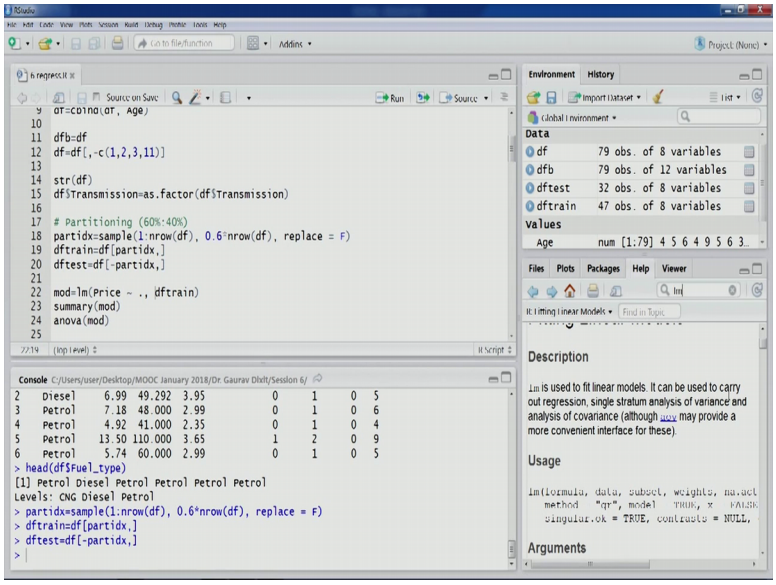
We will have to relabeled this variable. So, some this kind of exercise we will do when we you know discuss a particular technique that is suitable for classification or used for classification. So, next thing that would be required, once we have done on all very you know variable transformation, we have checked the variable type appropriately transform them, now all the variables are ready and we are ready for the modeling exercise. So, before we go ahead we need to partition on this particular sample. So, in this particular exercise will partition this particular dataset into 60 percent for the training set and the 40 percent for the test set.

So, we would not be having validation set we would be building our model on the training set and then we would be testing our model on this test partition. So, sample is the function that could be used to perform this partitioning. So, in the first argument as we have discussed before, we need to specify in a numeric vector form number of observations. So, in this case 1 2 number of rows that are there in this particular data frame representing the number of observations that are there in the data frame. So, that being indicated then the size of the sample, that is being indicated by this 0.6 into the length of the data frame.

So, because we want 60 percent of the observation 2 randomly selected observations to go into our training partition. So, therefore, 0.6 into this the full size, and the replacement is false because we want to do our sampling without replacement which is the typical way of sampling in a modeling exercise.

So, this particular sample function would return as the indexes indices of those observation which have been randomly drawn; now to further partition once. So, let us execute this line. So, you would see that part i d x this index variable has been created this is integer because these are the indices.

(Refer Slide Time: 22:09)



That have been returned by the sample function and you would see that 60 percent of the other observation, the indices of 60 percent of the observation randomly drawn from that particular data set have been written.

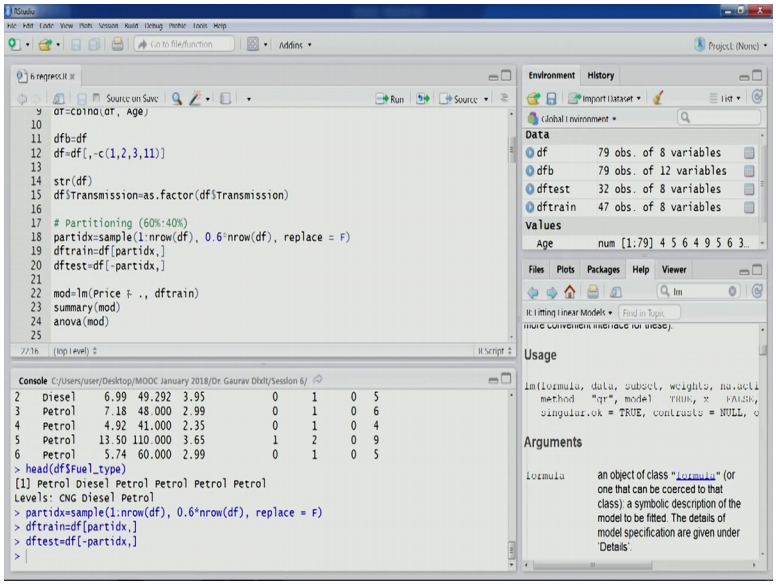
Now, we need to partition the data set. So, we can do using these brackets functions, we can subset these particular observations from the full data set. So, in the rows value in the for the row value within the brackets, we can mention this particular variable and all those indices would then be selected subsetted for training partitions. So, let us execute this code. So, we have been able to create d f train, you can see forty seven observations of 8 variables, which is same as the part i d x which was also 47 have which had 47 indices in the first place. Now since we are getting just 2 partitions therefore, all the remaining observation can go through can actually be left for the test partition.

So, in within the brackets in the for the row value we can mention minus part i d x. So, the remaining indices they would be subsetted for the test partition d f test. So, let us execute this code.

Now, once the partitioning exercise is over, now you can see in the environment section d f test another partition has been created having 32 observations of 8 variables. Now once this partitioning exercises has been done the function we can move to our linear modeling linear regression modeling. So, the function that is available in r for the to perform this modeling is l m. If you are interested in finding more details about this further function you can go to the help section and type l m and enter and you would see that in the help section it talks about this particular function, l m is about fitting linear models.

So, you can see in the description that it talks about that l m is used to fit linear models; it can be used to carry out regression, single stratum analysis of variance analysis of covariance right. So, all those statistical tests can be performed using this function.

(Refer Slide Time: 24:33)

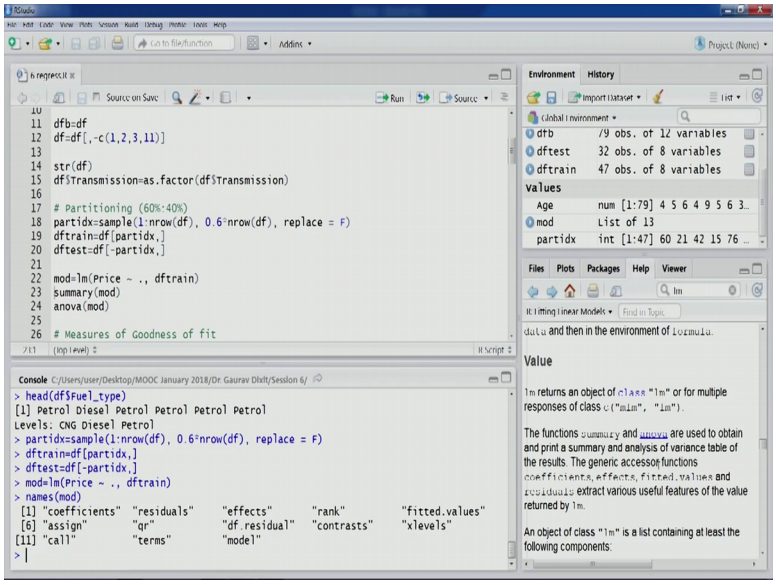


Now, if you look at the usage, you can look at the function and the arguments that can be fast first one is a formula, formula that is going to represent the linear regression model right. So, we need to pass this particular formula and then the second important variable is data. So, there are many other arguments also. So, you can on your own time you can go through some of these variables which are not typically used.

So, the formula is written in this particular format. So, the output outcome variable of interest it is written first and then we use the tilde operator and then we can write the all the names of the predictors that we have in our data set, and that and that we want the variables predictors that know which we want to include in our model. Or we can simply type dot if we want to include all the variables that are that are available in the data set. So, if you remember then the data frame that we are using now the that we partitioned, we had already excluded the variables that it we did not want in the first place and therefore, all the remaining variables are the come into our set of predictors and we would like to have all of them in our model.

So, our formula is going to be priced tilde dot; dot indicating to this function that all the all other variables should be part of the set of predictors. Now the data set that we are using is d f train that is the training partition. So, we would be building this model on training partition.

(Refer Slide Time: 26:14)

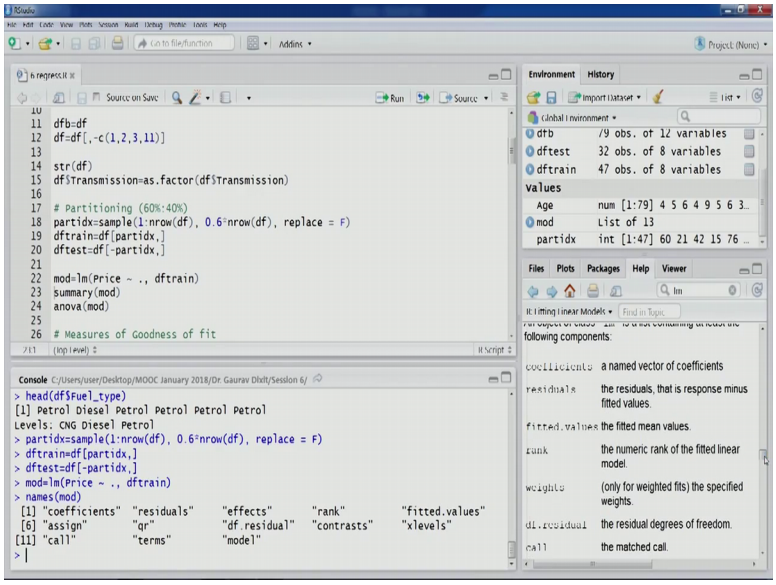


So, let us execute this line and you would see that in the environment section a mod variable has been created if you are not so, it has it is this this particular variable is actually a list of 3. So, it has information on thirteen elements, if you are interested in finding out the all the names of these this particular list, you can the names come on.

So, these are the thirteen elements that are there you can see. The first one is about coefficients, then second one is about residuals then effects, rank, fitted values, assignment you know similarly so many other details have been computed by this particular function.

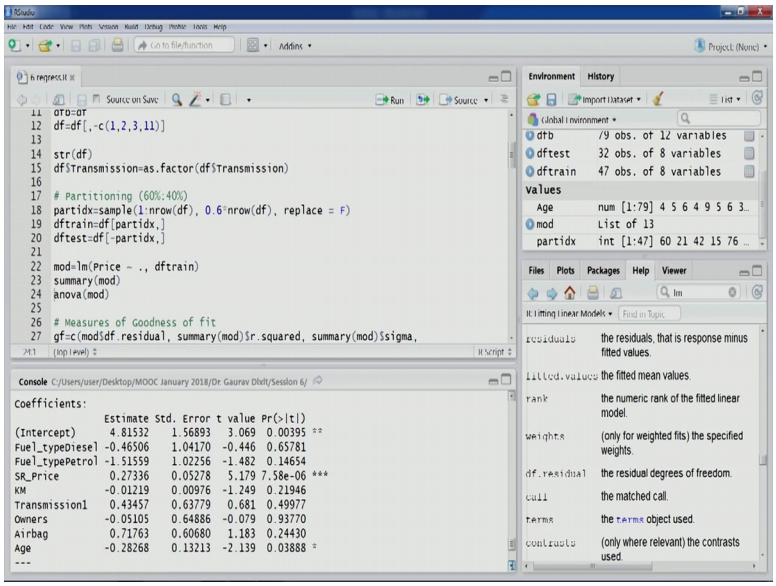
If you are understand finding out all these values, you can again go to the help section and find out and it is scroll down in this particular section, and you would see that there is going to be discussion under the value sub section. The kind of values that are returned by l m function and within this you will have details what are coefficients a named vector of coefficients of the coefficient that we have residuals fitted values.

(Refer Slide Time: 27:21)



So, details about all these written values would be there. So, we might not be interested in all these written values. So, somebody is one important function for the for us to get the relevant output from this exercise. So, let us execute this.

(Refer Slide Time: 27:57)



So, let us look at the summary output. So, you can see that the call explaining the formula that we had given there that we have the arguments that we are given to the l m function, then we have residuals some descriptive statistics about the residuals you can see that a minimum value median value residual, that this particular residual residual is of this models are having min max medium first quartile third quartile.

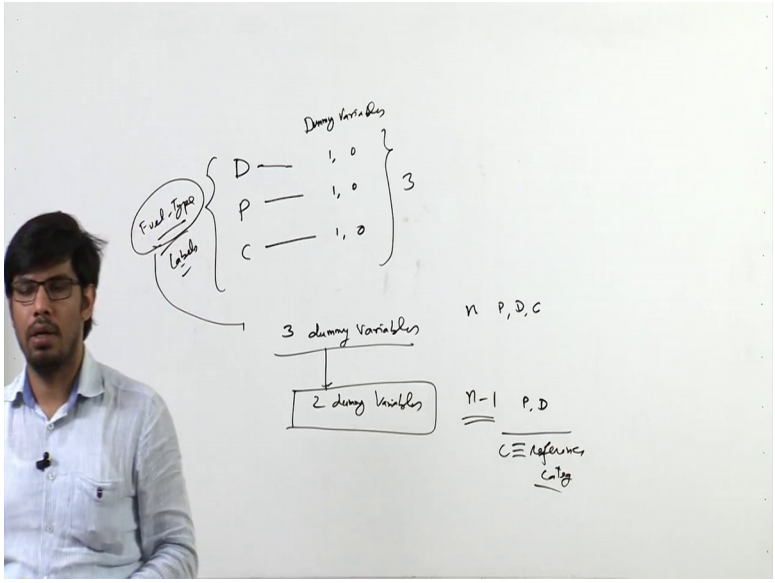
Now, let us come to the important part that we would like to discuss first coefficients; the coefficients you can see for all the predictors or we can see here first one is the intercept that is the constant. So, this particular estimator is representing the beta 0 that we had in our slide.

So, this particular intercept against intercept this particular value that we have this is beta 0, the corresponding standard error for this particular estimate is also given there. The t value and p value have the same meaning which we described in our supplementary lecture on introduction to basic statistics right. So, those values remain same.

We have also discussed a few more details about these values there. So, you can watch those particular lectures. Now the next important variable that we can see is fuel type. Now you would see that instead of one we have 2 variables for fuel type fuel type diesel and fuel type petrol why this has happened is, because fuel type is a categorical variable and it had 3 categories. So, for all those categories because the way software's are implemented, the way these techniques are implemented, they cannot handle the textual data and therefore, dummy codes dummy coding has to be performed on these variables categorical variables, and depending on the number of categories in a categorical variable we will have to create equal number of dummy variables.

So, dummy variables are actually representing different categories. So, for example, fuel type we had 3 categories diesel, petrol and CNG.

(Refer Slide Time: 30:18)



So, for each of these categories right diesel patrol and CNG. So, we had these 3 labels for fuel type for fuel type labels, for each of these labels will have to create dummy variables. So, dummy variables would indicate presence or absence of that particular value that particular label. So, if the value is one; that means, that car is having fuel type of diesel if the value is 0; that means, that car is not having fuel type of diesel.

Similarly, the variable the dummy variable corresponding dummy variable for petrol having value 1 will mean that the car is running on the fuel type of petrol, and if it is 0 it is not running on fuel type petrol similarly for this one. So, therefore, for each label will have to create equal number of dummy variable. So, we will have 3 variables. So, for the fuel type instead of having one variable in our model we will end up with 3 dummy variables to represent this particular categorical variable. But if we look at the variables presence of if you know if we have information on 2 variables, any 2 variables out of 3; if we have these any 2 variables out of these 3 then the other variables information is automatically known.

So, therefore, in our modeling exercise right if a particular value is if a particular car is not having diesel or petrol then of course, it will have CNG. So, because of that if we have information on 2 dummy variables the third one is automatically known. So, therefore, in our model we just have to include 2 dummy variables. So, if there are n classes then we will have to include n minus one dummy variables in the model. Now what happens to do the remaining label right.

So, for example, petrol and diesel are selected here, and we had petrol diesel and CNG. So, the remaining label it becomes the reference category that we have been talking about. So, any results that we get for these 2 variables now these 2 dummy variables p and d, those results would have to be interpreted with respect to this res reference category. More on this will stop here more on this particular dummy coding we will discuss in our next lecture.

Thank you.