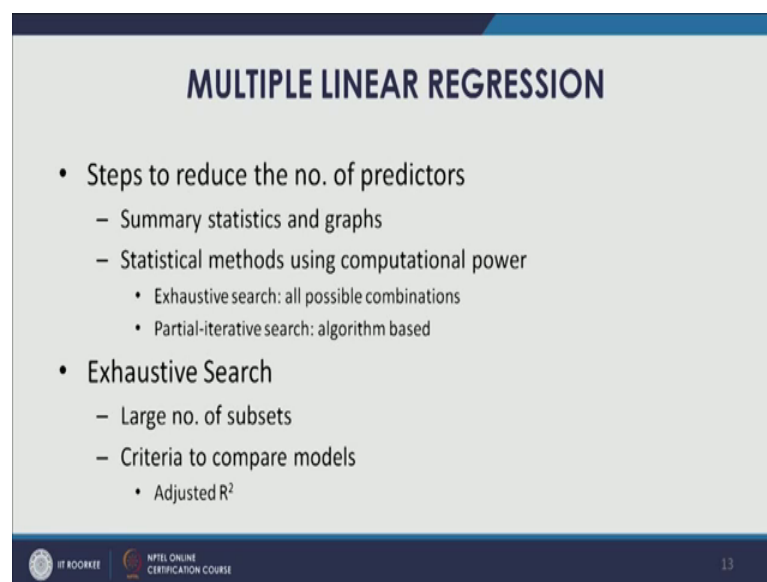


Business Analytics & Data Mining Modeling Using R
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture - 26
Multiple Linear Regression-Part V Exhaustive Search

Welcome to the course business analytics and data mining modeling using R. So, in the previous lecture we were discussing multiple linear regressions. And specifically we started our discussion on exhaustive search. So, there are many regression-based search algorithms that could be used to reduce the dimension or for variable selections as we have been discussing in the previous lecture as well.

(Refer Slide Time: 00:46)



MULTIPLE LINEAR REGRESSION

- Steps to reduce the no. of predictors
 - Summary statistics and graphs
 - Statistical methods using computational power
 - Exhaustive search: all possible combinations
 - Partial-iterative search: algorithm based
- Exhaustive Search
 - Large no. of subsets
 - Criteria to compare models
 - Adjusted R^2

IT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE | 13

So, let us start our discussion with exhaustive search. So, exhaustive search is about when we try all possible combinations of predictors. So, we are looking to examine each possible each possible combination of predictors, and check you know compare their performances and find out the best subsets from those possible combinations.

So, essentially, we are dealing with large number of subsets. So, there are a different criteria to compare model's subset models. So, the criterias 2 of them are same, as we do for any regression model R square and adjusted R square. So, first you know let us discuss adjusted R square before we proceed.

(Refer Slide Time: 01:43)

MULTIPLE LINEAR REGRESSION

- Adjusted R^2
$$R^2_{adj} = 1 - \frac{n-1}{n-p-1} (1 - R^2)$$

Where R^2 is proportion of explained variability in the model

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$
$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$
- R^2 is called coefficient of determination

IIT ROORKEE NPTEL ONLINE CERTIFICATION COURSE 14

So, adjusted R square can be defined using this particular formula this particular expression $1 - (n-1 / n - p - 1)$; where p is the number of predictors and n is the number of observation, multiplied by $(1 - R^2)$ that is p multiple R square, where R square is a proportion of explained variability in the model. So, R square is something that we have been using and will discuss a bit more on R square as well, R square is also called coefficient of determination. And mainly used in statistical modeling where we are generally looking for goodness of fit measures. So, therefore, we are trying to understand how much of the variability is being explained by the model, and R square being the matrix for the same.

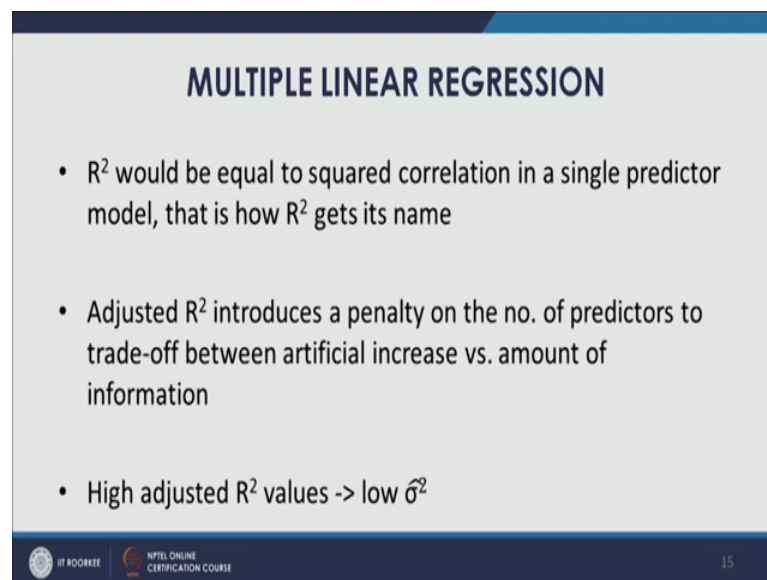
Now, adjusted R square in a way we can say the improved version of R square, and where we actually account for the degrees of freedom in a sense the number of predictors. So, our R is adjusted R square generally includes a penalty for number of predictors, thereby you would see that adjusted R squared value is always slightly less than the corresponding R square value. So, that is mainly because of the penalty that has been added due to the number of predictors.

So, more the predictors more penalty and that has to be accounted for. So, even look at the R squares definition or formula as well R square can be computed as following $1 - \text{SSE} / \text{SST}$, that is SSE is sum of squared deviations of sum of squares of errors, and SST is total sum of squares. you can also express this in the following form as

well 1 minus summation over i from 1 to n number of observation, and then y_i that is the actual value minus the predicted value that is $y_i - \hat{y}_i$, and then you can divide this particular numerator by summation over i from 1 to n , and then within the parentheses you can have the actual value y_i minus divided by the mean of this particular y .

So, in this fashion also you can compute R square value. Now R square is called as the resource coefficient of determination, and mainly used to check the goodness of fit. So, let us move further; now R square another way to understand R square is that it would be equal to a squared correlation in a single predictor model. So, if we just had a single predictor we just had y and that is being regressed on x_1 .

(Refer Slide Time: 04:32)



MULTIPLE LINEAR REGRESSION

- R^2 would be equal to squared correlation in a single predictor model, that is how R^2 gets its name
- Adjusted R^2 introduces a penalty on the no. of predictors to trade-off between artificial increase vs. amount of information
- High adjusted R^2 values \rightarrow low $\hat{\sigma}^2$

IT ROOKIE | NPTEL ONLINE CERTIFICATION COURSE | 15

So, if we just had y and x_1 and we are looking for a linear regression model for the same. Then the R square would be squared correlation that being the single predicted case. And that is how our square also gets its name. So, in that case the correlation coefficient generally is expressed using a small R and if you square that R square. So, that is how in a single predator case R square gets its name coefficient of determination. Now as we as I discussed adjusted R square introduces a penalty on the number of predators to trade-off between artificial increased verses amount of information. So, that this is the trade-off that is generally incorporated and adjusted R square value.

So, if there are more number of predictors and they are uncorrelated to the outcome variable. thereby they would just artificially increasing they would just be artificially increasing the R square value, and would not be contributing much to the model in terms of amount of information. So, the R square adjusted R square will consider this trade off, and will impose and penalty for the same.

So, we do not want artificial increase in R square right. So, we want some amount of information some contribution coming from those predators information to the model in terms of explaining the variability increasing the variability that is in the outcome variable that is being explained by the model. So, if we look at a few more things for example, high adjusted R square values. So, that would also mean that we will get a lower sigma square low sigma hat square.

So, that is also so, we are eventually we will get a low variance. So, high adjusted R square values, and that would also indicate this. Now another great area to compare model models that could be used in exhaustive research is mallow's C p.



(Refer Slide Time: 06:45)

MULTIPLE LINEAR REGRESSION

- Exhaustive Search
 - Criteria to compare models
 - Mallow's C_p
- Mallow's C_p

$$C_p = \frac{SSR}{\hat{\sigma}_f^2} + 2(p + 1) - n$$

Where $\hat{\sigma}_f^2$ is estimated value of σ^2 in the full model
and $SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$

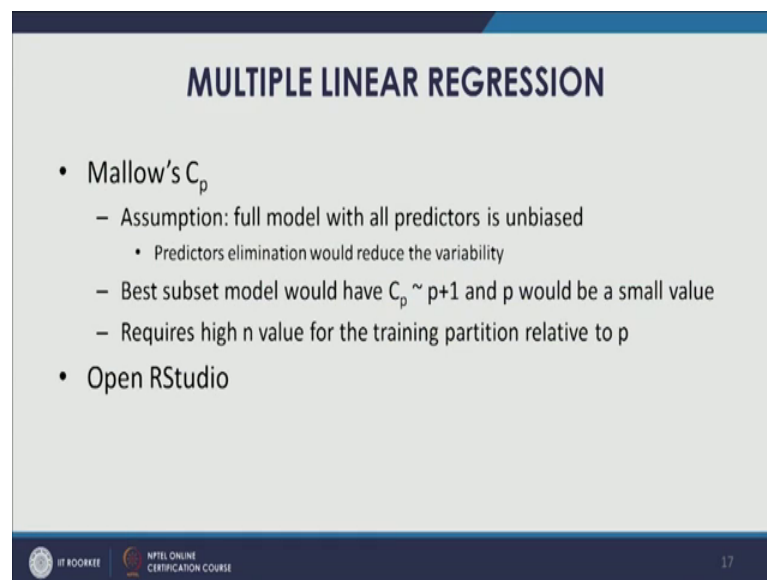
 IIT KHARAGPUR
  NPTEL ONLINE CERTIFICATION COURSE
 16

Mallow's C p is can be expressed in this form C p SSR that is sum of a squares sum of squares for regression. And then divided by this sigma for full model sigma hat full model square, then twice p plus 1 minus n. So, this being the how we compute mellow C p, now sigma f hat square is estimated value of sigma square in the full model. And SSR

is as expressed here is the y hat the predicted value minus that mean value of y . So, that is the sum of squares for regression; so, this is what we have.

So, mallow C_p can also be used to compare models, how it can be used? We will just discuss. So, one assumption that you can see in mallow C_p is that full model assumption is that full model with all predictors is unbiased.

(Refer Slide Time: 07:37)



MULTIPLE LINEAR REGRESSION

- Mallows' C_p
 - Assumption: full model with all predictors is unbiased
 - Predictors elimination would reduce the variability
 - Best subset model would have $C_p \sim p+1$ and p would be a small value
 - Requires high n value for the training partition relative to p
- Open RStudio

17

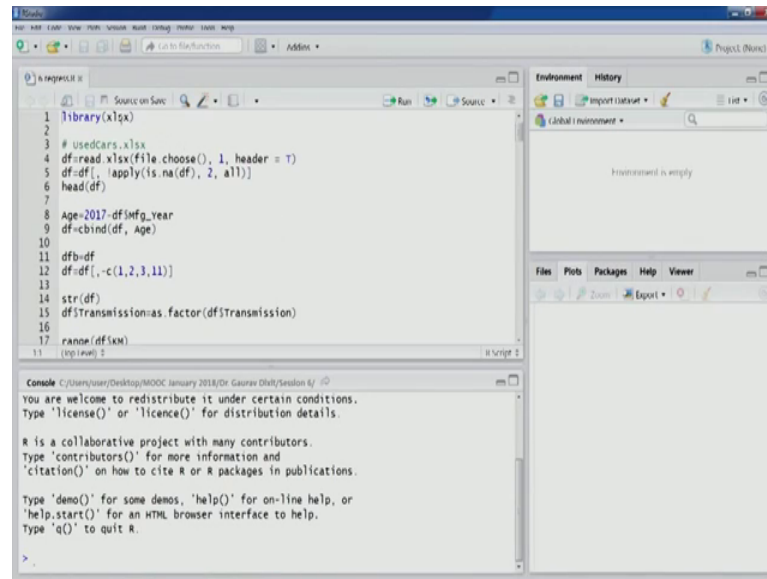
So, that is the assumption that we make, full model with all predictors. So, that is how we start when we are talking about exhaustive research. So, we would we would also be because we are would be exploring all possible combination of combinations of predictors. So, therefore, this would also full model will also be considered. So, assumption is that full model with all predictors is unbiased right. So, that is then that would also mean that predictors elimination would reduce the variability.

So, if we eliminate predictors. So, that it is going to reduce the variability and that is the desirable thing for us. So, how do we find out using this particular criterion? How do we find out the best subset model? So, best subset model would have C_p closure to p plus 1 value and p would be a small value. So, C_p value that we compute using the formula that we just saw in the previous slide. So, it would be closer to p plus 1 and the p would be a small value. So, using these 2 using these 2 criteria's criteria we can actually find out the best subset model now. another effect another important point related to mallow C_p is that it requires high n value more number of observation for the training partition

related to p. So, depending on the number of arrive because, we would we considered considering all possible combinations, though therefore, with respect to p, we would be requiring more number of observations in the training partitions.

Now, let us open R studio, and we will understand these concept through an exercise.

(Refer Slide Time: 09:43)

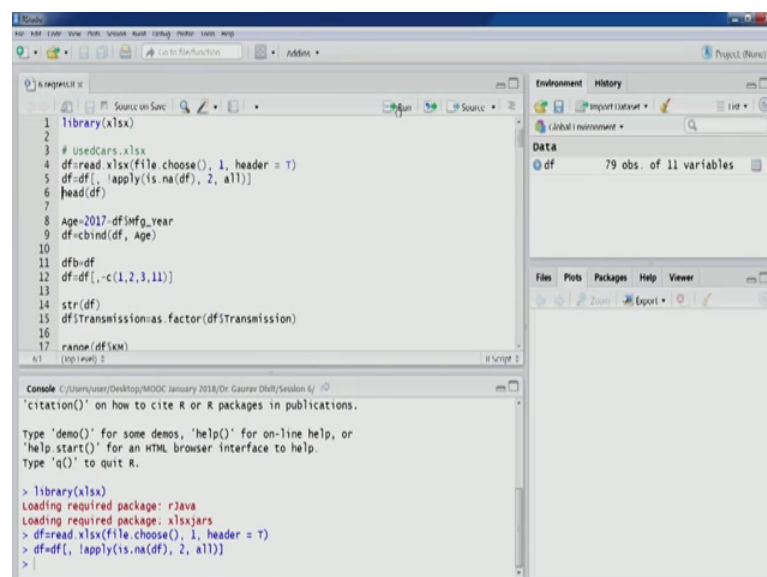


```
1 library(xlsx)
2
3 # UsedCars.xlsx
4 df=read.xlsx(file.choose(), 1, header = T)
5 df=df[, !apply(is.na(df), 2, all)]
6 head(df)
7
8 Age=2017-df$Mfg_Year
9 df=cbind(df, Age)
10
11 dfb=df
12 df=df[,c(1,2,3,11)]
13
14 str(df)
15 df$Transmission=as.factor(df$Transmission)
16
17 range(df$km)
```

The console shows the R startup message: "You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details. R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications. Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R."

So, as usual let us load this particular library, the data set that again we are using.

(Refer Slide Time: 09:49)



```
1 library(xlsx)
2
3 # UsedCars.xlsx
4 df=read.xlsx(file.choose(), 1, header = T)
5 df=df[, !apply(is.na(df), 2, all)]
6 head(df)
7
8 Age=2017-df$Mfg_Year
9 df=cbind(df, Age)
10
11 dfb=df
12 df=df[,c(1,2,3,11)]
13
14 str(df)
15 df$Transmission=as.factor(df$Transmission)
16
17 range(df$km)
```

The console shows the execution progress: "Loading required package: rJava", "Loading required package: xlsxjars", and the execution of the data loading and manipulation code.

The Environment pane shows the data frame 'df' with 79 observations and 11 variables.

Used cars data set let us import this. So, this is the file let us import can see 79 observation of 11 variables in the environment section and allow let us remove any columns. And let us look at the first 6 observation we are already familiar with this particular dataset.

(Refer Slide Time: 10:11)

```

1 library(xlsx)
2
3 # UsedCars.xlsx
4 df=read.xlsx(file.choose(), 1, header = T)
5 df=df[, !apply(is.na(df), 2, all)]
6 head(df)
7
8 Age=2017-df$Mfg_Year
9 df=cbind(df, Age)
10
11 dfb=df
12 df=df[, -c(1,2,3,11)]
13
14 str(df)
15 df$Transmission=as.factor(df$Transmission)
16
17 rannoe(df$KM)

```

Console output:

```

> library(xlsx)
> df=read.xlsx(file.choose(), 1, header = T)
> df=df[, !apply(is.na(df), 2, all)]
> head(df)
  Brand Model Mfg_Year Fuel_Type SR_Price KM Price Transmission Owners
1 Hyundai Verna    2013   Petrol    8.88  75.000  5.60         0      1
2 Mahindra Quanto  2012   Diesel    6.99  49.292  3.95         0      1
3 Maruti Suzuki SX4  2011   Petrol    7.18  48.000  2.99         0      1
4 Chevrolet Beat   2013   Petrol    4.92  41.000  2.35         0      1
5 Honda Civic      2008   Petrol   13.50 110.000  3.65         1      2
6 Honda Brio       2012   Petrol    5.74  60.000  2.99         0      1

```

Environment pane shows: Data df, 79 obs. of 11 variables.

So, again you can see brand model manufacturing year fuel type SR price KM price being the outcome variable of interest, and others being the predictors transmission owners airbag and c price.

(Refer Slide Time: 10:20)

```

1 library(xlsx)
2
3 # UsedCars.xlsx
4 df=read.xlsx(file.choose(), 1, header = T)
5 df=df[, !apply(is.na(df), 2, all)]
6 head(df)
7
8 Age=2017-df$Mfg_Year
9 df=cbind(df, Age)
10
11 dfb=df
12 df=df[, -c(1,2,3,11)]
13
14 str(df)
15 df$Transmission=as.factor(df$Transmission)
16
17 rannoe(df$KM)

```

Console output (continued):

```

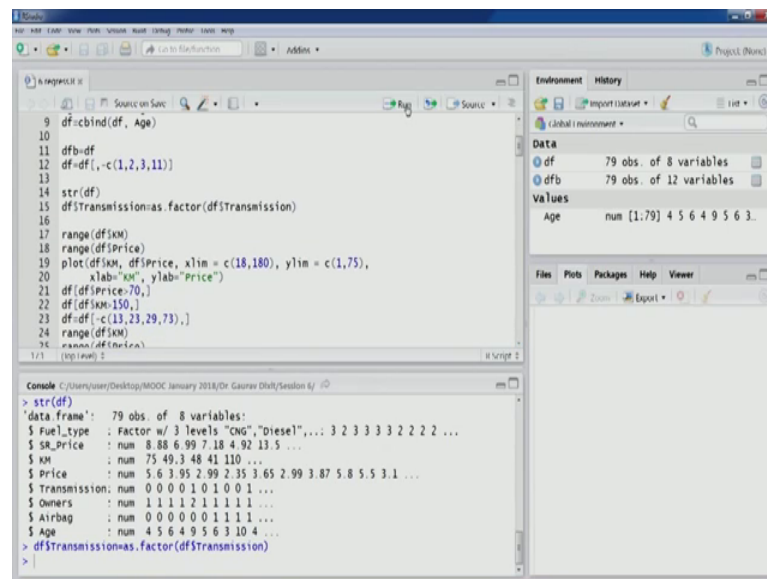
6 Honda Brio 2012 Petrol 5.74 60.000 2.99 0 1
Airbag C_Price
1 0 1
2 0 0
3 0 0
4 0 0
5 0 0
6 0 0
> Age=2017-df$Mfg_Year
> df=cbind(df, Age)
> dfb=df
>

```

Environment pane shows: Data df, 79 obs. of 12 variables. Values: Age num [1:79] 4 5 6 4 9 5 6 3..

So, right now we are not interested in c price. So, that would also be removed. So, first thing that we will do is compute this particular variable age using manufacturing year column, and that is append this to data frame let us take a backup. And then we would be getting rid of first c variables which are not off interest to us and the another one c underscore price as well.

(Refer Slide Time: 10:51)



The screenshot shows the RStudio interface. The main editor window contains the following R code:

```
9 df=cbind(df, Age)
10
11 dfb=df
12 df=df[,c(1,2,3,11)]
13
14 str(df)
15 df$Transmission=as.factor(df$Transmission)
16
17 range(df$KM)
18 range(df$Price)
19 plot(df$KM, df$Price, xlim = c(18,180), ylim = c(1,75),
20      xlab="KM", ylab="Price")
21 df[df$Price>70,]
22 df[df$KM>150,]
23 df=df[-c(13,23,29,73),]
24 range(df$KM)
25 range(df$Price)
```

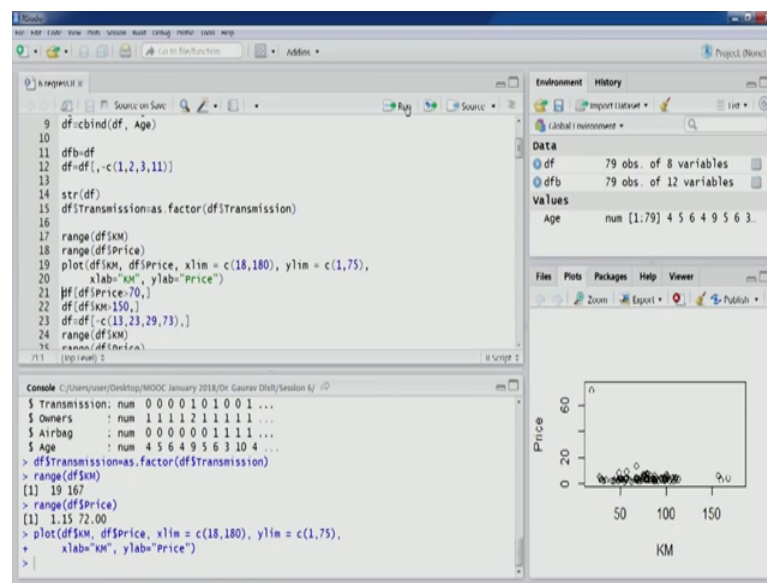
The console window at the bottom shows the output of the `str(df)` command:

```
> str(df)
'data.frame': 79 obs. of 8 variables:
 $ Fuel_type : Factor w/ 3 levels "CNG","Diesel",...: 3 2 3 3 3 3 2 2 2 ...
 $ SR_Price : num 8.88 6.99 7.18 4.92 13.5 ...
 $ KM : num 75 49.3 48 41 110 ...
 $ Price : num 5.6 3.95 2.99 2.35 3.65 2.99 3.87 5.8 5.5 3.1 ...
 $ Transmission: num 0 0 0 0 1 0 1 0 0 1 ...
 $ Owners : num 1 1 1 1 2 1 1 1 1 1 ...
 $ Airbag : num 0 0 0 0 0 1 1 1 1 ...
 $ Age : num 4 5 6 4 9 5 6 3 10 4 ...
> df$Transmission=as.factor(df$Transmission)
```

The Environment pane on the right shows the objects `df` and `dfb`, both with 79 observations and 8 variables. The 'Values' section shows the range of the 'Age' variable: `num [1:79] 4 5 6 4 9 5 6 3`.

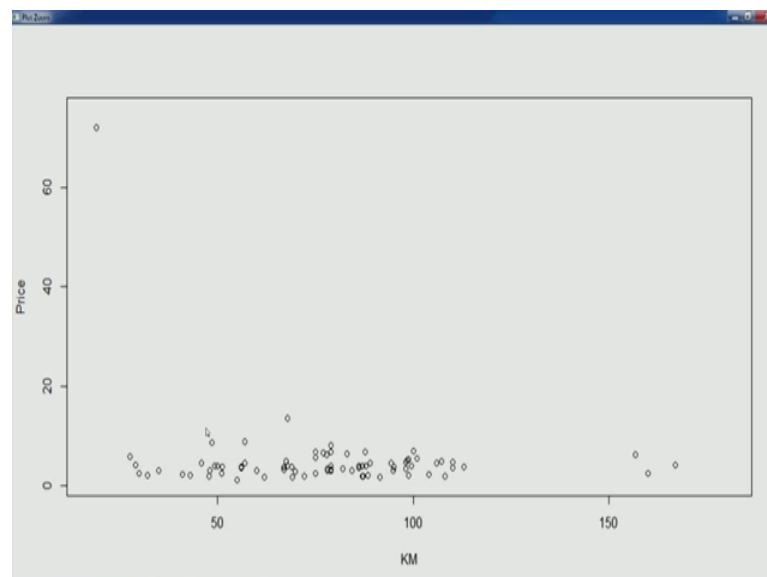
So, now we can look at the structure of this particular data frame you can see that 79 observation of 8 variables now, and all the variables of interest here are in this data frame. Now be a transmission is also a categorical variable automatic or manual the 2 variables are using you know coded using numeric codes. So, let us also convert this into a factor variable using this particular code. Now let us also after this let us also plot this KM versus price scatter plot.

(Refer Slide Time: 11:30)



You would see now from this particular plot you can see there are few outliers very clear outliers.

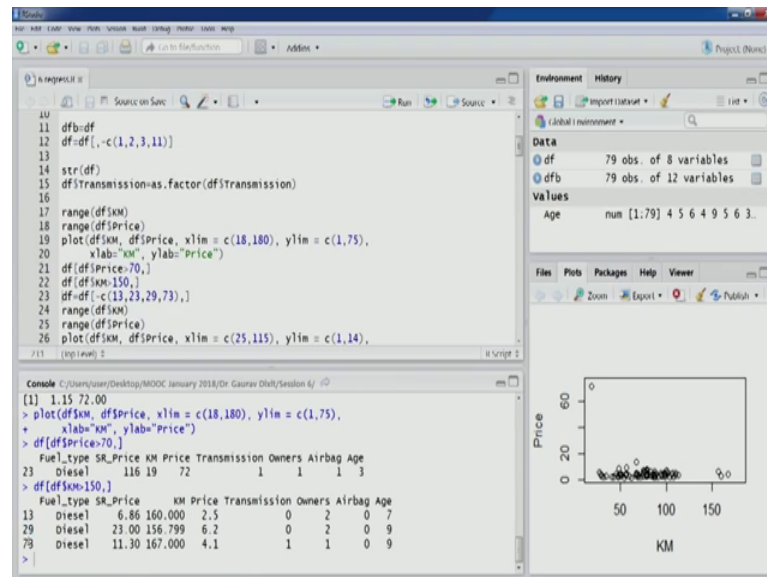
(Refer Slide Time: 11:38)



So, majority of the majority of the values they are lying in this particular zone, right, between 0 to 120 on y axis, and somewhere between 0 to 100, 20, 30 in on x axis, very small part of this particular plot where the values are lying.

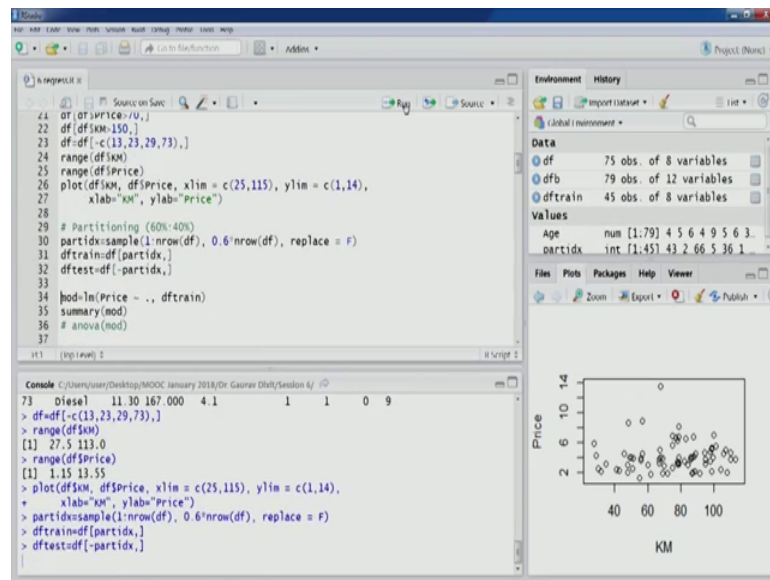
So, other values seemed clearly to be outlier values. So, let us get rid of them on y axis direction, you can see that values greater than 70 there was one particular point, greater than 70 this is the point surprise having 100 and the price having 72.

(Refer Slide Time: 12:13)



And then similarly on the x axis that is for the kilometre in the x axis direction, we have 3 values greater than 150. So, we can identify those rows as well. So, you can see what we are trying to identify is the row indexes for all these values. So, these 4 are large we have been able to identify the indices for all of them. And once identified we can use the brackets to subset this particular data set. So, once we achieve this code, and combination function combine function and then we would be left with only 75 observation. You can see from 79 we have dropped to 75 observation of 8 variables.

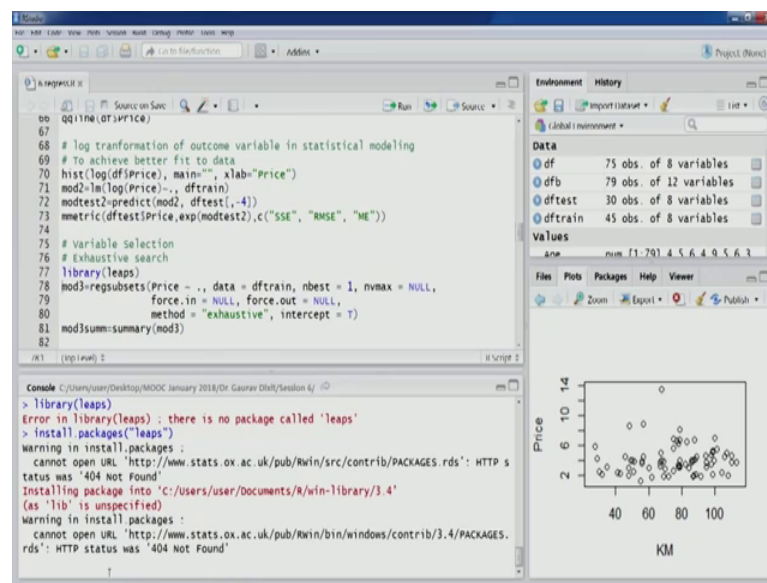
(Refer Slide Time: 13:00)



Now, again we can plot and see how this is the scatter plot has changed between price and kilometre and now you can see most of the plotting region is being occupied by the points. And all the points are close by no outlier seems to be there. Now as usual we will do the partitioning.

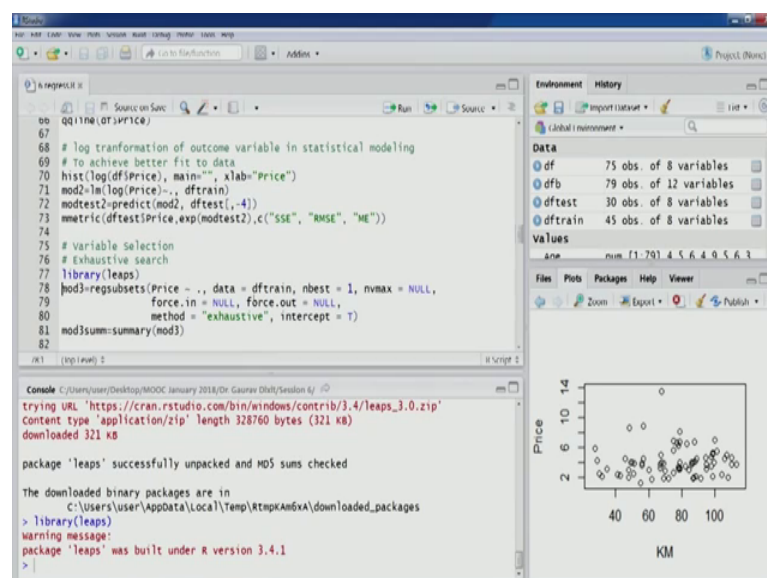
Once partitioning is done so, we will directly skip to the part where we can discuss exhaustive search. So, let us skip to that part, this is the part, variable selection, and exhaustive search is the first method that we are going to discuss. So, library leaps is the leaps is the library that we need to load for this particular method exhaustive search. And reg subsets. So, regression-based subsets is the function, and that we would be using. So, let us load this particular library. So, this does not seem to be installed so, let us first install this particular package.

(Refer Slide Time: 14:05)



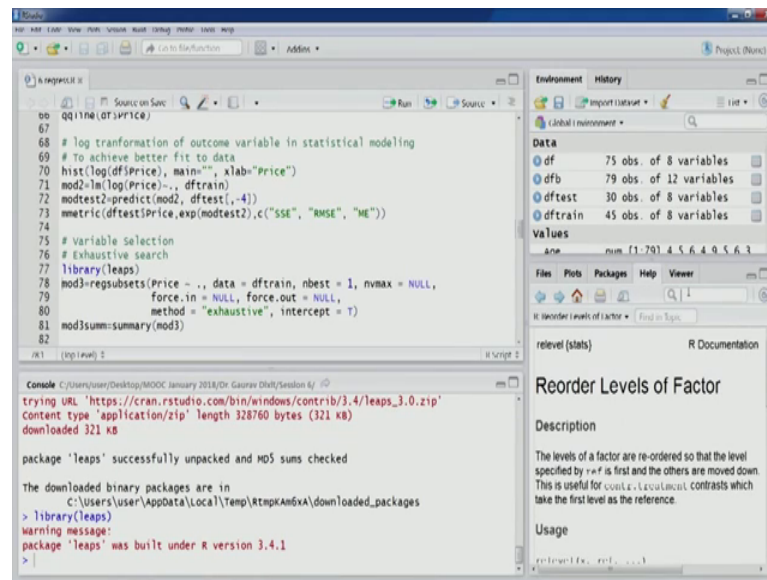
As we have discussed before for installing a particular package, you just have to pass it as an argument in the installed dot packages function, and once you do this and if you have internet connection this will start downloading the required packages and also install then let us reload the library once now that this is installed so, what is being loaded.

(Refer Slide Time: 14:35)



Now, we would be able to use this function, reg subsets we are interested in finding more information about this program function.

(Refer Slide Time: 14:39)



```
67 qq1line(df$PRICE)
68 # log transformation of outcome variable in statistical modeling
69 # To achieve better fit to data
70 hist(log(df$Price), main="", xlab="Price")
71 mod2=lm(log(Price)~., data=dftrain)
72 modtest2=predict(mod2, dftrain[,4])
73 mmetric(dftrain$Price,exp(modtest2),c("sse", "RMSE", "ME"))
74
75 # Variable Selection
76 # Exhaustive search
77 library(leaps)
78 mod3=regsubsets(Price ~ ., data = dftrain, nbest = 1, nvmax = NULL,
79 force.in = NULL, force.out = NULL,
80 method = "exhaustive", intercept = T)
81 mod3sum=summary(mod3)
82
```

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/leaps_3.0.zip'
Content type 'application/zip' length 328760 bytes (321 kB)
downloaded 321 kB

package 'leaps' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
c:\users\user\AppData\Local\Temp\rtmpkAm6xA\downloaded_packages

```
> library(leaps)
Warning message:
package 'leaps' was built under R version 3.4.1
>
```

Environment: History

Data

- df 75 obs. of 8 variables
- dfb 79 obs. of 12 variables
- dftest 30 obs. of 8 variables
- dftrain 45 obs. of 8 variables

Values

Price num [1, 79] 4 5 6 4 9 5 6 3

Files Plots Packages Help Viewer

Reorder Levels of Factor

Description

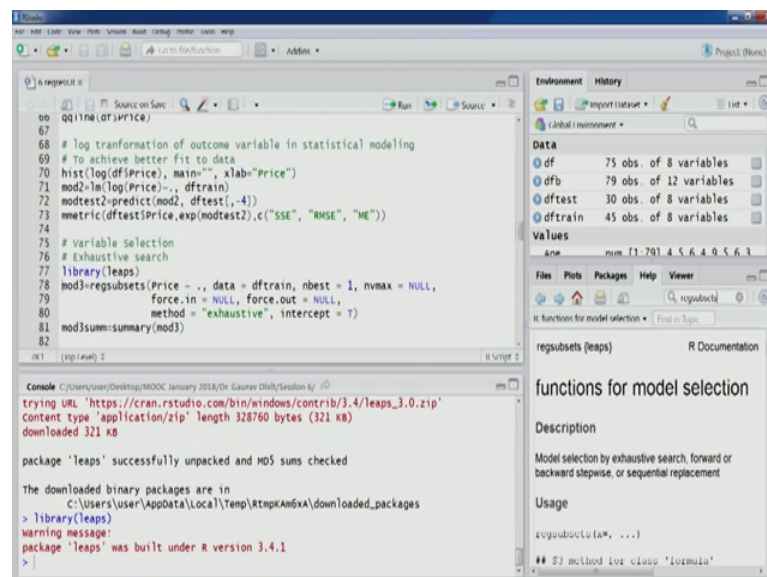
The levels of a factor are re-ordered so that the level specified by `ref` is first and the others are moved down. This is useful for `contr.treatment` contrasts which take the first level as the reference.

Usage

```
reorder(fx, ref, ...)
```

You can go into the help section reg subests, and you would be able to find out more information on this, functions for model selection.

(Refer Slide Time: 14:48)



```
67 qq1line(df$PRICE)
68 # log transformation of outcome variable in statistical modeling
69 # To achieve better fit to data
70 hist(log(df$Price), main="", xlab="Price")
71 mod2=lm(log(Price)~., data=dftrain)
72 modtest2=predict(mod2, dftrain[,4])
73 mmetric(dftrain$Price,exp(modtest2),c("sse", "RMSE", "ME"))
74
75 # Variable Selection
76 # Exhaustive search
77 library(leaps)
78 mod3=regsubsets(Price ~ ., data = dftrain, nbest = 1, nvmax = NULL,
79 force.in = NULL, force.out = NULL,
80 method = "exhaustive", intercept = T)
81 mod3sum=summary(mod3)
82
```

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/leaps_3.0.zip'
Content type 'application/zip' length 328760 bytes (321 kB)
downloaded 321 kB

package 'leaps' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
c:\users\user\AppData\Local\Temp\rtmpkAm6xA\downloaded_packages

```
> library(leaps)
Warning message:
package 'leaps' was built under R version 3.4.1
>
```

Environment: History

Data

- df 75 obs. of 8 variables
- dfb 79 obs. of 12 variables
- dftest 30 obs. of 8 variables
- dftrain 45 obs. of 8 variables

Values

Price num [1, 79] 4 5 6 4 9 5 6 3

Files Plots Packages Help Viewer

regsubsets (leaps)

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(x[, ...])
## $3 method for class 'lmAmla'
```

So, in this case variable selection and few other algorithms are covered for example, model selection by exhaustive search forward or backward step wise or sequential replacement. So, these are the methods that are supported by this particular function.

So, in this function as you can see that first we have to express the formula because essentially this is a regression based. So, a regression models would be built. So, price is

there were our outcome variable of interest, and the dot representing the other predictors in the data set, but as part of the, this exhausted search various combinations of various pairs of combinations of predictors are going to be tried data is d f train and other arguments if you can find out from the help section. So, we have appro appropriately specified all those arguments. The most important being the method where we have selected exhaustive as the method for our exercise. So, once this is done.

(Refer Slide Time: 15:55)

```

70 m1st=log(OT3PRICE), m1st2=log(OT3PRICE)
71 mod2=lm(log(Price)~., dftest)
72 modtest2=predict(mod2, dftest[,4])
73 mmetric(dftest1Price,exp(modtest2),c("SSE", "RMSE", "ME"))
74
75 # Variable Selection
76 # Exhaustive search
77 library(leaps)
78 mod3=regsubsets(Price ~ ., data = dftest, nbest = 1, nvmax = NULL,
79               force.in = NULL, force.out = NULL,
80               method = "exhaustive", intercept = T)
81 mod3sum=summary(mod3)
82
83 countspsch=function(x) sum(x=="*")
84 om=as.integer(apply(mod3sum$outmat, 2, countspsch)); om
85 data.frame("coeffs"=as.integer(apply(mod3sum$outmat, 1, countspsch)),
86           "RSS"=mod3sum$rss,

```

Environment

- dftest 79 obs. of 12 variables
- dftest2 30 obs. of 8 variables
- dftest1 45 obs. of 8 variables
- Age num [1:79] 4 5 6 4 9 5 6 3
- mod3 List of 28
- mod3sum List of 8

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(x[, ...])
```

S3 method for class 'formula'

```
regsubsets(x, data, weights=NULL, nbest=1000, nvmax=NULL, force.in=NULL, force.out=NULL, method="exhaustive", intercept=T)
```

Console

```

C:\Users\user\Desktop\MOOC January 2018\Dr. Gaurav Dhillon\session 4\
package 'leaps' successfully unpacked and MD5 sums checked

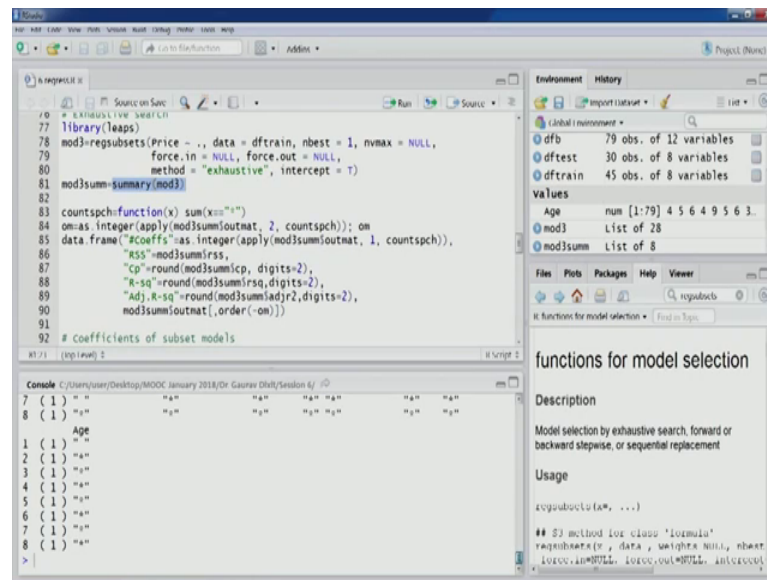
The downloaded binary packages are in
C:\Users\user\AppData\Local\Temp\RtmpKAm6XA\downloaded_packages
> library(leaps)
Warning message:
package 'leaps' was built under R version 3.4.1
> mod3=regsubsets(Price ~ ., data = dftest, nbest = 1, nvmax = NULL,
+               force.in = NULL, force.out = NULL,
+               method = "exhaustive", intercept = T)
> mod3sum=summary(mod3)

```

So, we can execute this particular code. And model has been build we can also find out the summary.

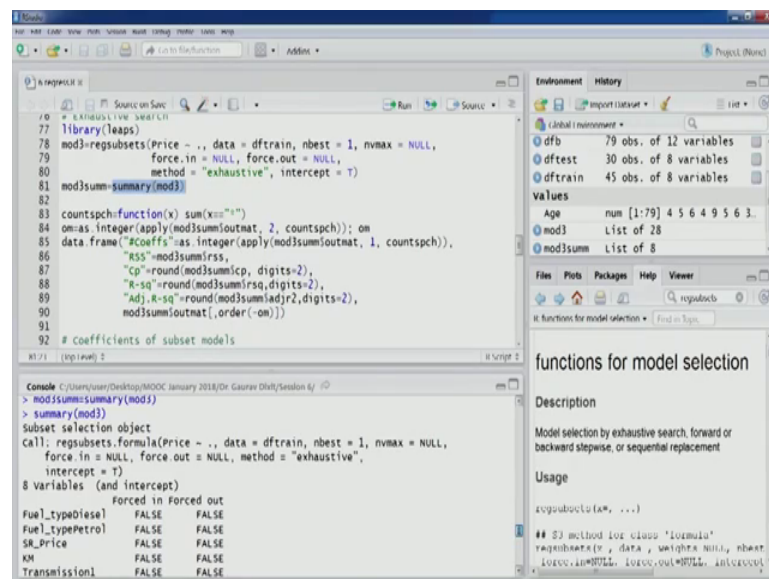
Now, in the in the in the in the summary output and the, and as we will see just a bit now; that there are going to be the variables, that would be counted using asterisk. So, first let us let me show you the summary here.

(Refer Slide Time: 16:21)



So, this is the summary result that we have. So, in this case you can see.

(Refer Slide Time: 16:27)



a call and the 8 variables and intercept that are there.

(Refer Slide Time: 16:30)

The screenshot shows the RStudio interface. The main editor displays the `mod3sum` function, which uses `regsubsets` to generate model subsets and `counts` to count the occurrences of each variable in the subsets. The console shows the output of the function, which is a matrix where rows represent model subsets (1 to 8) and columns represent variables. Asterisks in the matrix indicate which variables are included in each subset. The right-hand pane shows the environment and history, listing objects like `dfb`, `dftrain`, and `mod3sum`.

```

70 library(leaps)
71 mod3=regsubsets(Price ~ ., data = dftrain, nbest = 1, nvmax = NULL,
72               force.in = NULL, force.out = NULL,
73               method = "exhaustive", intercept = T)
74 mod3sum=summary(mod3)
75 counts=counts(mod3sum$outmat, 2, counts=TRUE)
76 om=as.integer(apply(mod3sum$outmat, 2, counts))
77 data.frame("coeffs"=as.integer(apply(mod3sum$outmat, 1, counts)),
78           "RSS"=mod3sum$RSS,
79           "Cp"=round(mod3sum$Cp, digits=2),
80           "R-sq"=round(mod3sum$Rsq, digits=2),
81           "Adj.R-sq"=round(mod3sum$AdjRsq, digits=2),
82           mod3sum$outmat[,order(-om)])
83 # coefficients of subset models

```

Console output:

```

Age FALSE FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
Fuel_typeDiesel Fuel_typePetrol SR_Price KM Transmission Owners Airbag
1 (1) " " " " " " " " " " " "
2 (1) " " " " " " " " " " " "
3 (1) " " " " " " " " " " " "
4 (1) " " " " " " " " " " " "
5 (1) " " " " " " " " " " " "
6 (1) " " " " " " " " " " " "
7 (1) " " " " " " " " " " " "
8 (1) " " " " " " " " " " " "

```

So, you would see in the results we have one subset of each side up to 8 SR price is selected as you can see asterisk indicating that a surprise is selected. We will discuss more of the results once we get produce an the output in the suitable format for us to analyze. So, let us do that so, these s take this function that I have written in the next line; count is special character this function is going to count the instances of a asterisk in a particular row or column right.

So, let us create this function. Now this function will be using to for counting for these as you can see the in the assembly output that we just saw for a particular column we would be counting these asterisk . So, that this counting will help us reorder the columns of this particular matrix.

So, this being the matrix we would like to reorder the columns probably SR price we would like to see first because as S R price seems to be present in all 8 models. So, therefore, and then followed by the next variable next column which comes in more rows right which appears in more rows therefore, we would like to order in that in that sense. And for that we need to count asterisk as part of the computation. This particular this particular out mat would actually have this output matrix.

So, we are trying to account the this apply function you are already familiar, the second argument to secured argument value of 2 indicating that we are going to apply this function on a column wise. And therefore, we will get the numbers of asterisk for each

column. So, let us compute this, you can see these numbers now these numbers are going to be passed on to, pass down as argument to this matrix itself when we construct a data frame for our output.

(Refer Slide Time: 18:34)

```

76 # EXHAUSTIVE SEARCH
77 library(leaps)
78 mod3-regsubsets(Price ~ ., data = dftrain, nbest = 1, nvmax = NULL,
79               force.in = NULL, force.out = NULL,
80               method = "exhaustive", intercept = T)
81 mod3sum=summary(mod3)
82
83 countspch=function(x) sum(x=="*")
84 om=as.integer(apply(mod3sum$outmat, 2, countspch)); om
85 data.frame("Coeffs"=as.integer(apply(mod3sum$outmat, 1, countspch)),
86          "RSS"=mod3sum$rss,
87          "Cp"=round(mod3sum$cp, digits=2),
88          "R-sq"=round(mod3sum$rsq, digits=2),
89          "Adj.R-sq"=round(mod3sum$adjr2, digits=2),
90          mod3sum$outmat[,order(-om)])
91 # coefficients of subset models
92

```

Console output:

```

1 (1) "*"
2 (1) "*"
3 (1) "*"
4 (1) "*"
5 (1) "*"
6 (1) "*"
7 (1) "*"
8 (1) "*"
> countspch=function(x) sum(x=="*")
> om=as.integer(apply(mod3sum$outmat, 2, countspch)); om
[1] 1 6 8 5 2 4 3 7

```

Environment pane:

- dftrain: 45 obs. of 8 variables
- mod3: List of 28
- mod3sum: List of 8

You can see ordered by here, you can see ordered by minus om which we have just computed.

So, the count the particular column having the most number of asterisk, we want it to appear first and then followed by the next column having more number of asterisk. So, that is why this arrangement this has been done. Now in the data of frame you would see the first column is number of coefficients. So, for each model we want to see number of coefficients that are there.

So, that can be again computed using this asterisk right. So, this does not count the intercept term. So, intercept is another that could be there so, we are not counting it. So, again you can see near apply function now we are counting row wise the second argument the apply function is one. So, therefore, we are counting row wise. Then we have this RSS which is residual sum of squares.

So, this is again in the output itself. So, we would be using this then the C p values we have we want to see we are interested in just the value up to 2 decimal points similarly R square and adjusted R square it is R square. So, the 3 criteria that we have to compare

different subset models models are 3 rather 4 criteria is RSS residual sum of square then values C p then R square and then adjusted R square. So, we would be using these statistics, these criteria's to compute different subsets models.

Last one last particular variable would is it is actually a matrix would actually indicating the, the presence of presence of or absence of different variables in the particular model. So, let us compute this, let us look at the output. So, you can see RSS value for all 8 models.

(Refer Slide Time: 20:46)

The screenshot shows the RStudio interface with the `modsums` package workflow. The script in the editor defines a function `countspsch` and uses `modsums::summary(mod3)` to generate a summary of subset models. The console output displays a table with columns for the number of variables (1 to 8), coefficients (X.coefs), RSS, Cp, R-squared, Adjusted R-squared, and the selected variables (SR_Price, Age, Fuel_type, Petrol, KM). The Adjusted R-squared value peaks at 0.72 for the 4-variable model.

	X.coefs	RSS	Cp	R.sq	Adj.R.sq	SR_Price	Age	Fuel_type	Petrol	KM
1 (1)	1	71.56760	32.29	0.49	0.48	*				
2 (1)	2	47.78747	9.94	0.66	0.64	*	*			
3 (1)	3	37.81514	1.73	0.73	0.71	*	*	*		
4 (1)	4	36.12397	1.99	0.74	0.72	*	*	*	*	
5 (1)	5	35.36094	3.21	0.75	0.71	*	*	*	*	*
6 (1)	6	35.18017	5.03	0.75	0.71	*	*	*	*	*
7 (1)	7	35.15838	7.00	0.75	0.70	*	*	*	*	*
8 (1)	8	35.15357	9.00	0.75	0.69	*	*	*	*	*

Owners Airbag Transmission Fuel_type Diesel

So, the first model is having one coefficient or one variable then the second model 2 variable 3 variables and in this fashion up to 8 variables that are there. Now you would see though we had 8 variables there is was one categorical variable. So, we had to dummy variable for the same. So, therefore, we had though we had 7 variables, it is showing as it just like the regression output, where we had 2 variables representing fuel type right fuel type diesel and petrol.

So, access values are there that is residual sum of squares and C p values are there a square and adjusted R square value are there. If we focus first on adjusted R square value, you can see starting from one variable model to 2 variable model, the R square value keeps increasing right, keeps increasing and when we reach to 4 variable model, that is this 4th row right when we reach to this particular 4 variable model the R square value has peaked at small value of 0.72.

And after that as we include more variables, 5 variable model 6 variable model, you would see that the R square adjusted R square value is decreasing right.

So, that is because the as we discussed before, that adjusted R square it imposes and penalty on number of predictors. So, therefore, because of that so, there is not that that to quantum increase in R square value, and the with respect to further increase in number of predictors is not good enough, and that is why penalty has been imposed and adjusted say R square value is then decreases after that after 4 variable model. But if we look at the R square column, you would see that these value keeps R square value keeps on increasing and 4 variable model it reaches it is peak that is 0.74, and then it increases to 0.75. And for more you know if we add more variable model we look at more 6 7 8 variable model, and then there also the value remains at same 0.75.

So, maybe it is increasing you know after 2 decimal points or 3 4 fifth decimal point we might see some increase. So, essentially, we can say R square value keeps on increasing. So, even though the predictor information might not be useful for the model in terms of contributing the, you know contributing as you know amount of information. But the uncorrelated even though the information even though the predictor might not be contributing in terms of information, but R square value keeps on increasing right. So, if we look at the adjusted R square value probably the 4-variable model for variable subset. That is the one that we would like to select similarly, if you look at the mallow's C p. So, as we discussed the Cp value a closer to p plus 1, and also, we can have a look at the lower p value.

So, if we look at this and specifically we start at the 3-variable model, you can see 1.73 and p value is at this point is 3. So, 3 plus 1 that is 4. So, 1.73 is and the differences from slightly more gap between 1.73 value and 4. We look at the next value for variable model, you can you can see 1.9, and that is about 2, and the number of variables now are 4 plus 1 5. So, that gap is 3 here, and the earlier gap was for 3 variable model 1.73 to 4 so, that was about 2 point something. So, 3 variable model 4 variable model. So, these are the models then if you look at the 5-variable model asterisk from mallow C p the value increases to 3.21.

Now, 3.21 and the p value is 5. So, that makes it 6 this is also. So, 5 variable model is also 2, 3, 4, 5. They both are in the that the cap is similar, right. If we move further then

the sheep mallow C p value is 5.03, and the p value is 6. So, that is comes out to be 6 plus 1 7. So, even less than value, but as we discussed that we are interested in low p value right. So, if we look at the C p allow C p, then probably will will select the 3-variable model, right. Because the difference is 1.73 and difference between 1.73 and 4 ah; that is, 2 point about 2.27, and, but the value of p is low that is 3, but if we compare it to the 5-variable model all, let us say 6 variable model.

So, in this case 6 variable model let us say 5, and that that difference is 7 the difference is less, but it it is higher p value 5 variable model 3.21. And p value is 5 so, 5 plus 1 6. So, the difference is still 2 point more than that. So, therefore, one the 4-variable model there also the difference is more than that, then probably using the C p value we look at it, then 3 variable model would be selected looking at the adjusted R square value, the 4-variable model would be selected.

And looking at the R square value 5 variable model would be selected because that is the highest R square that we can have, and after that we just keep on being if we just look at the 2 decimal points. So, after that it is just the number of variables keeps added and the R square value is increasing after 2 decimal points the third 4th decimal point there might be some increase.

Now if we look at the variables that are included in these models, you can see the first column, and that is why we had ordered this particular matrix in terms of the number of asterisk that are present. Because, you immediately we can identify that a SR price being the most important variable, in a sense that it is appearing in all 8 models followed closely by age which is also appearing in 7 of these models.

Then fuel type petrol which is appearing in 5 of these 6 of these models, then kilometre is appearing in 5 of these models, all right. So, in that sense, we can generate this you know in a very importance of we can also understand the importance of variables that is S R by price is appearing in all the models definitely the most important ; which is also expected that because we are trying to predict the used price of a car and the show show room price being the main indicator it is not surprising.

But let me also tell you as I have indicated before that, this particular analysis that we are doing is based on just 75 observation very small data set therefore, there is it is subject to the results are subject to change to partitioning that we are doing. So, every time when

we do a partitioning, and when we change it the results might change significantly. So, therefore, if we do this same exercise using a much larger data set then probably, even if we repeat the exercise that is a small change to that extent, right.

Now if we are interested in few more information for example, coefficient of subsets models, for example, second there is this coefficient function it is going to give us the coefficient value for our different subset models that we have just computed eight models.

(Refer Slide Time: 28:23)

The screenshot shows the RStudio environment. The script editor contains the following code:

```

79 force.in = NULL, force.out = NULL,
80 method = "exhaustive", intercept = 1)
81 mod3sum=summary(mod3)
82
83 countspch=function(x) sum(x=="*")
84 om=as.integer(apply(mod3sumloutmat, 2, countspch)); om
85 data.frame("#Coeffs"=as.integer(apply(mod3sumloutmat, 1, countspch)),
86           "RSS"=mod3sumlrss,
87           "Cp"=round(mod3sumlcp, digits=2),
88           "R-sq"=round(mod3sumlrq, digits=2),
89           "Adj. R-sq"=round(mod3sumlajr2, digits=2),
90           mod3sumloutmat[,order(-om)])
91
92 # coefficients of subset models
93 # second argument is index vector indicating ordering in summary output
94 coef(mod3, 1:8)
95

```

The Environment pane shows the following objects:

- dftrain: 45 obs. of 8 variables
- dfptest: 30 obs. of 8 variables
- mod3: List of 28
- mod3sum: List of 8
- om: int [1:8] 1 6 8 5 2 4 3 7

The Console shows the output of the code:

```

7 ( 1 )      7 35.15838      7.00 0.75      0.70      * *
8 ( 1 )      8 35.15357      9.00 0.75      0.69      * *
      Owners Airbag Transmission1 Fuel_typeDiesel
1 ( 1 )
2 ( 1 )
3 ( 1 )
4 ( 1 )
5 ( 1 )
6 ( 1 )
7 ( 1 )
8 ( 1 )

```

The right pane shows the documentation for the `regsubsets` function, including a description and usage.

So, the second argument is actually a index for the same the output that we saw in the summary and otherwise the data frame that we constructed. So, 1 2 8 8 models are there. So, let us look at the coefficient values. So, let us start from the first. So, you can see the first model when we look at the one variable model, then the model is actually based on a surprise the one predictor; that is, there is a surprise coefficient value we can also .2 4 6.

(Refer Slide Time: 28:59)

```

83 countspec=function(x, sum, k=1) {
84   om=as.integer(apply(mod3sumoutmat, 2, countspec)); om
85   data.frame("Rcoeffs"=as.integer(apply(mod3sumoutmat, 1, countspec)),
86             "RSS"=mod3sumirss,
87             "Cp"=round(mod3sumicp, digits=2),
88             "R-sq"=round(mod3sumirsq, digits=2),
89             "Adj.R-sq"=round(mod3sumiadjr2, digits=2),
90             mod3sumoutmat[,order(-om)])
91
92 # Coefficients of subset models
93 # second argument is index vector indicating ordering in summary output
94 coef(mod3, 1:8)
95
96 # Partial, iterative search
97 # Forward selection
98 mod4=regsubsets(Price~., data = dftrain, nbset = 1, mvmax = NULL,
99               force.in = NULL, force.out = NULL,

```

```

> coef(mod3, 1:8)
[[1]]
(Intercept)  SR_Price
1.7981689    0.2459522

[[2]]
(Intercept)  SR_Price  Age
3.5977612    0.2802512 -0.3656525

[[3]]
(Intercept) Fuel_typePetrol  SR_Price  Age

```

And then if you look at 2 variable model than S R price, and age are there help me move forward than fuel type petrol S R price. And age are there and then fuel type petrol S R price KM and age and as we move further the 5-variable model.

(Refer Slide Time: 29:30)

```

83 countspec=function(x, sum, k=1) {
84   om=as.integer(apply(mod3sumoutmat, 2, countspec)); om
85   data.frame("Rcoeffs"=as.integer(apply(mod3sumoutmat, 1, countspec)),
86             "RSS"=mod3sumirss,
87             "Cp"=round(mod3sumicp, digits=2),
88             "R-sq"=round(mod3sumirsq, digits=2),
89             "Adj.R-sq"=round(mod3sumiadjr2, digits=2),
90             mod3sumoutmat[,order(-om)])
91
92 # Coefficients of subset models
93 # second argument is index vector indicating ordering in summary output
94 coef(mod3, 1:8)
95
96 # Partial, iterative search
97 # Forward selection
98 mod4=regsubsets(Price~., data = dftrain, nbset = 1, mvmax = NULL,
99               force.in = NULL, force.out = NULL,

```

```

[[4]]
(Intercept) Fuel_typePetrol  SR_Price  KM  Age
4.56626891   -1.07571364    0.26705794 -0.01122816 -0.30720072

[[5]]
(Intercept) Fuel_typePetrol  SR_Price  KM  Owners
4.1971710   -1.1809765    0.2706734  -0.0127651  0.5188228
Age
-0.3208142

[[6]]

```

Then we see that owners is also there, and then 6 variable model and then now so, if you if one airbag has also appeared now. So, in this fashion you can see the coefficient value and which variables depending on the variable models the that. That is 5 variable model for 4 variable models 6 variable model. Will we recall the results that the R square

adjusted R square value using adjusted R square value using 0 on 4 variable model. We look at the that we take adjusted R square as the primary criteria for our exercise. Then four variable model we can see the variables that are there the fuel type petrol S R price KM and age.

So, probably these are the important variable which are contributing to some extent, and which is reflected and this for example, show room price is definitely important for predicting price of a you know used car kilometres. kilometre is also important, you can see you can also see that this is negatively negatively correlated here, and then you can look at the age also which is also negatively and rightly and you get it correlated.

So, that is also there you can also see at fuel type petrol is also negatively correlated; which is also true in the sense that diesel cost they are slightly you know they carry more price the show room at you know when first purchase at the time of first purchase they carry more price. And even for used cars it is the you know when this as per this data this petrol is negatively.

So, diesel or CNG with respect to because this reference is CNG so, with respect to the CNG petrol is a negatively priced. So, with this decision will stop here. And in the next next lecture, we will discuss some of partial iterative search algorithms for variable selection.

Thank you.