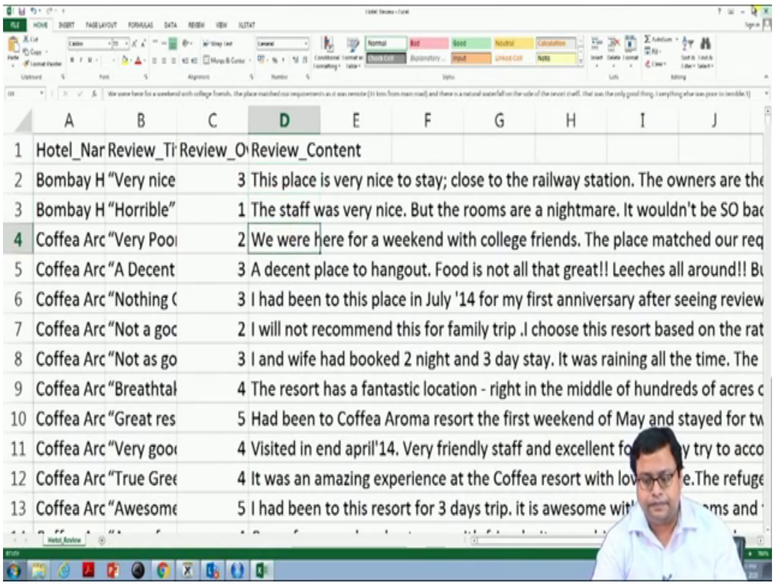
Marketing Analytics
Professor Swagato Chatterjee
Vinod Gupta School of Management,
Indian Institute of Technology, Kharagpur
Lecture 56
Text Mining and Sentiment Analytics (Contd.)

Hello everybody. Welcome to Marketing Analytics course, this is Doctor Swagato Chatterjee from VGSOM, IIT Kharagpur who is taking this course and we are in week 11 and we are still discussing Text Mining and Sentiment Analysis and we will continue the same thing in this particular week also. Till the last week we have done basic sentiment mining, basic text mining and we have also done some bit of spam detection and etcetera using Naive Bayes algorithm here we will actually use that in sentiment mining.

So we will use Naive Bayes for sentiment mining in this session. In the next session we will use certain libraries which has been pre-developed by some researchers for various kinds of application in various places and later point of time we will do a little bit of topic modeling also. So in this particular class we will be using the dataset called hotel_review. So this is the same old dataset which have been used earlier by us. But here I have only included the text part.

(Refer Slide Time: 01:24)



	A	B	C	D
1	Hotel_Nar	Review_Ti	Review_O	Review_Content
2	Bombay H	“Very nice	3	This place is very nice to stay; close to the railway station. The owners are the
3	Bombay H	“Horrible”	1	The staff was very nice. But the rooms are a nightmare. It wouldn't be SO bac
4	Coffea Arc	“Very Poo	2	We were here for a weekend with college friends. The place matched our req
5	Coffea Arc	“A Decent	3	A decent place to hangout. Food is not all that great!! Leeches all around!! Bu
6	Coffea Arc	“Nothing (	3	I had been to this place in July '14 for my first anniversary after seeing review
7	Coffea Arc	“Not a goc	2	I will not recommend this for family trip .I choose this resort based on the rat
8	Coffea Arc	“Not as go	3	I and wife had booked 2 night and 3 day stay. It was raining all the time. The
9	Coffea Arc	“Breathtal	4	The resort has a fantastic location - right in the middle of hundreds of acres c
10	Coffea Arc	“Great res	5	Had been to Coffea Aroma resort the first weekend of May and stayed for tw
11	Coffea Arc	“Very gooi	4	Visited in end april'14. Very friendly staff and excellent fo y try to acco
12	Coffea Arc	“True Gree	4	It was an amazing experience at the Coffea resort with lo e.The refuge
13	Coffea Arc	“Awesome	5	I had been to this resort for 3 days trip. it is awesome wit ms and

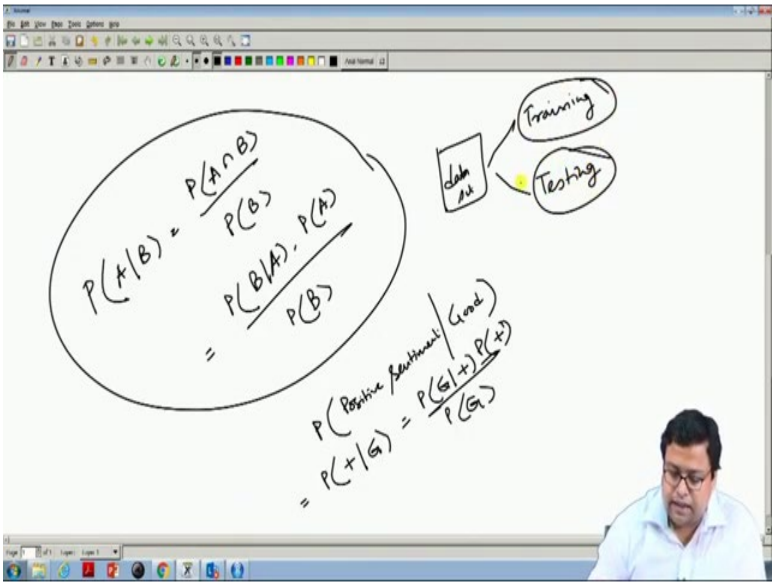
So if you see this dataset carefully that this dataset has 4 columns which is the hotel name, the review title, the overall rating that you have got and the review content the basic text of the review and these are all content like this place is very nice to stay close to the railway station. The owners are the base and will help you with all your questions.

We would definitely recommend this place blah, blah, blah all of these things, so each of this thing is one review. So the quantitative version of this particular thing, the dataset we have used in week 1 if you remember. In the week 1 we started with a discussion that if the overall rating is there and if you also get let us say the service, location, service, quality you get the ratings of all those stuff.

Then how various service attributes are impacting your ultimate overall satisfaction or overall rating that you have given is sometimes we find out through regression. We also try to find out that which of the aspects are more important in your service context, which aspects are not important, where you are doing good, where you are doing bad and blah, blah, blah. The same dataset we have been using here, but with the text part.

So I want to know that what kind of extra insights we can bring in using the text also till now we are only using quantitative data now I am bringing in qualitative data in this particular model building to find out that how we can get enough information from the qualitative data also to make certain sense out of it. So in this particular class we will do that. So we will be using as usual R programming and the Naive Bayes algorithm to do that.

(Refer Slide Time: 03:21)


$P(A|B) = \frac{P(A \cap B)}{P(B)}$
$= \frac{P(B|A) \cdot P(A)}{P(B)}$
$=$
data set
Training
Testing
$P(\text{Positive Sentiment} | \text{Good})$
$= P(+|G) = \frac{P(G|+) P(+)}{P(G)}$

So if you remember a small recap of the Naive Bayes algorithm it says that

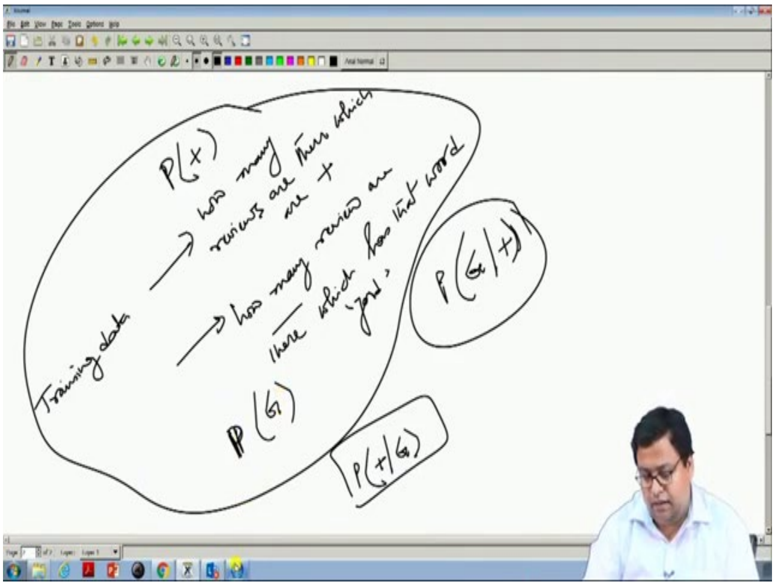
$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A).P(A)}{P(B)}$$

This is something that we will be using.

For example probability of positive sentiment let us say positive sentiment given a word called good.

Let us say if that is so probability of this given G, that is I have write it in this way, this will be P(G|+)/P(G). Now in our old dataset if I have a dataset and if I break into training and testing, there in a training data from the training data I can have an idea that.

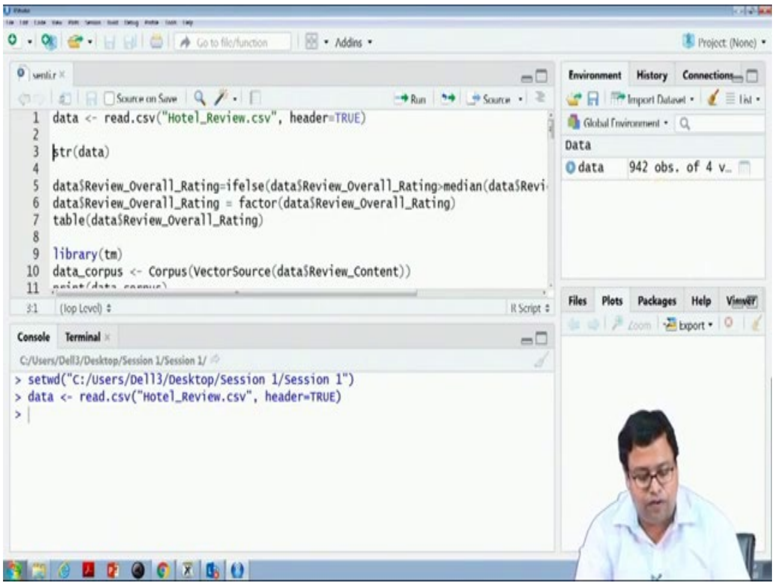
(Refer Slide Time: 04:25)


Training data
P(+)
how many reviews are there which are +
how many reviews are there which has that word 'good'
P(G)
P(G|+)
P(+|G)

In the training data I can have an idea that how many I would say reviews are there which are positive and I can also get a idea that how many reviews are there which has this word has the word called good. So this will be basically probability of plus and probability of G P(G|+) and then we have to find out by doing another calculation what is this. We can find out this thing then I can calculate probability of positive given G.P(+|G).

So these 3 simple data mining will be done by Naive Bayes algorithm. So the first thing is that so these are P actually I have written it in a different way this is it should be P, okay. So that is something that we will be doing here.

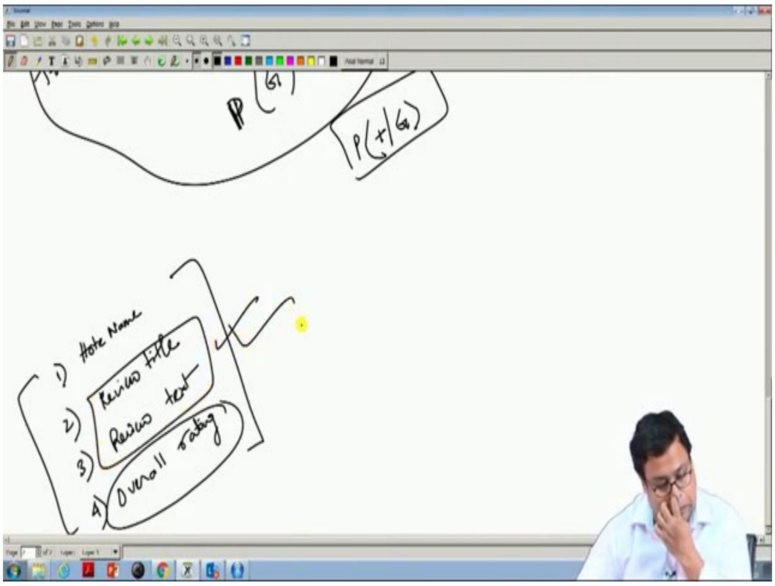
(Refer Slide Time: 05:42)



```r
data <- read.csv("Hotel_Review.csv", header=TRUE)

str(data)

data$Review_Overall_Rating=ifelse(data$Review_Overall_Rating>median(data$Revi
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
```

Data
data 942 obs. of 4 v...

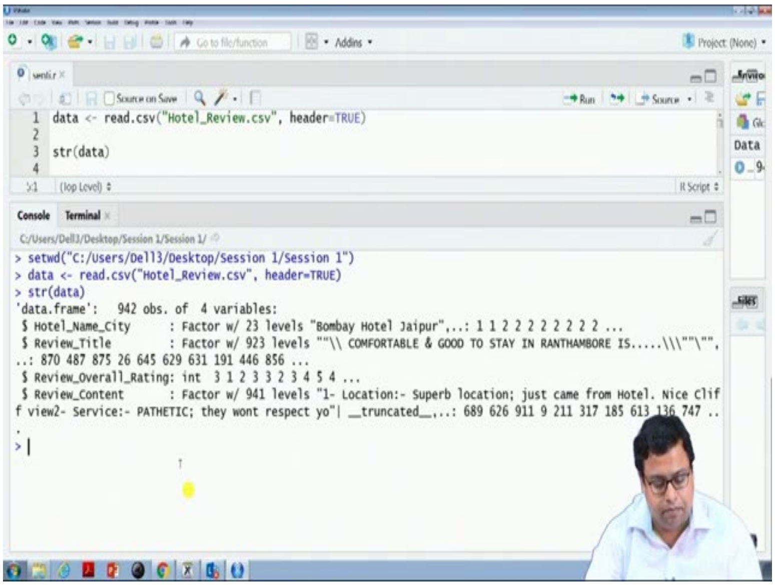
```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> setwd("C:/Users/Dell3/Desktop/Session 1/Session 1")
> data <- read.csv("Hotel_Review.csv", header=TRUE)
>
```



So if you check this particular Senti.R this is the R file. So we are using the same old formula of Naive Bayes, but in this context in this particular context which is our review dataset we will be using this thing. So the first job is to read the data. So session, set working directory to source file location and then after setting the working directory you read the data and the data has 942 observations of 4 variables.

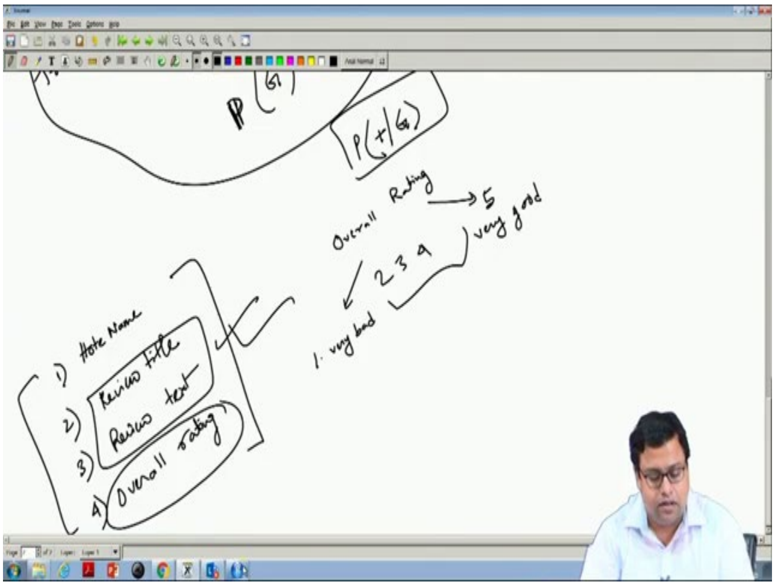
So how the data looks like? The data has the hotel name, the data has the review title, data has the review text and the data also has overall ratings. So these are the 4 things that our dataset has and we will try to see that how this overall rating is related to this two stuff that if you title and then if you text.

(Refer Slide Time: 06:52)



```
data <- read.csv("Hotel_Review.csv", header=TRUE)

str(data)
```

```
> setwd("C:/Users/Dell3/Desktop/Session 1/Session 1")
> data <- read.csv("Hotel_Review.csv", header=TRUE)
> str(data)
'data.frame':   942 obs. of  4 variables:
 $ Hotel_Name_City      : Factor w/ 23 levels "Bombay Hotel Jaipur",..: 1 1 2 2 2 2 2 2 2 2 ...
 $ Review_Title         : Factor w/ 923 levels ""\\ COMFORTABLE & GOOD TO STAY IN RANTHAMBORE IS.....\\\""\"",
..: 870 487 875 26 645 629 631 191 446 856 ...
 $ Review_Overall_Rating: int  3 1 2 3 3 2 3 4 5 4 ...
 $ Review_Content       : Factor w/ 941 levels "1- Location:- Superb location; just came from Hotel. Nice Clif
f view2- Service:- PATHETIC; they wont respect yo"| __truncated__,..: 689 626 911 9 211 317 185 613 136 747 ..
.
> |
```



Now if we come to this thing first thing that is we want to do is structure of the data. So the structure of the data says that my dataset has basically if you carefully look at the structure then my dataset has 942 observations of 4 variables there are 4 variables and there are 942 observations means 942 rows and 4 columns are there and the columns names are hotel name city which is a factor variable with 23 levels.

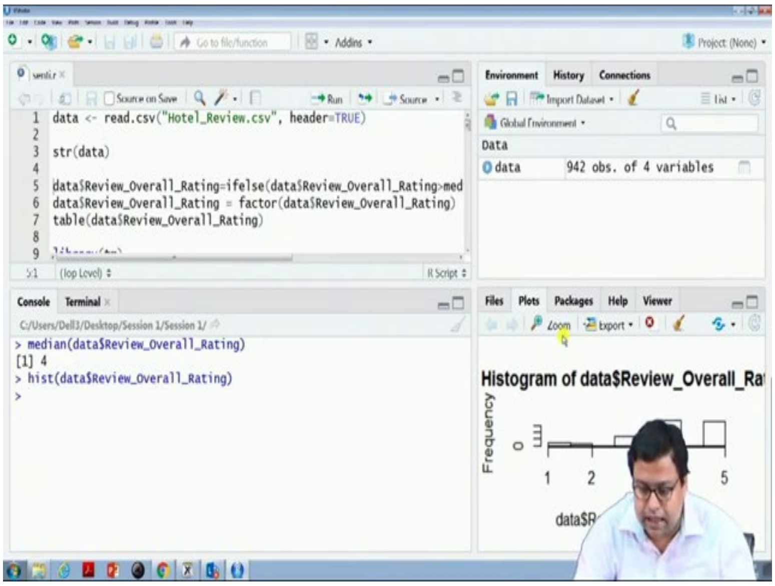
That means there are 23 different hotels data has been captured in this particular dataset and the review title has, review title is a factor of 923 levels that basically there are 942 different kinds of review titles are there it is coming 923 because some review titles are probably same let us good stay has come more than once. So if that is the case then we cannot say that they are different review title.

There can be some common review title for example very good hotel let us say that was the review title, so we cannot do anything with that. Then comes the review overall rating. Review overall rating is the quantitative rating in a 1 to 5 point scale that somebody has given 1 means in this overall rating 1 means basically very bad and 5 means very good and anything in between that let us say 2, 3, 4 these are mediocre.

So depending on how much is the cutoff we will decide whether it is a positive review or negative review and then comes review content which is the text and there are 941 levels because probably one thing what there is exactly two reviews which are similar to each other otherwise they are dissimilar to each other. So we got basically these 942 observations. Now I just press control L to clean my console control L is to clean my console.

And once I clean the console next what will I do. So what I do right now is I break the dataset the currently given dataset into two halves. One is the positive reviews and another is the negative reviews. So (what do I) how to find out positive review? We call it something called median split. What is median split? Find out the median of a variable and then split it.

(Refer Slide Time: 09:23)

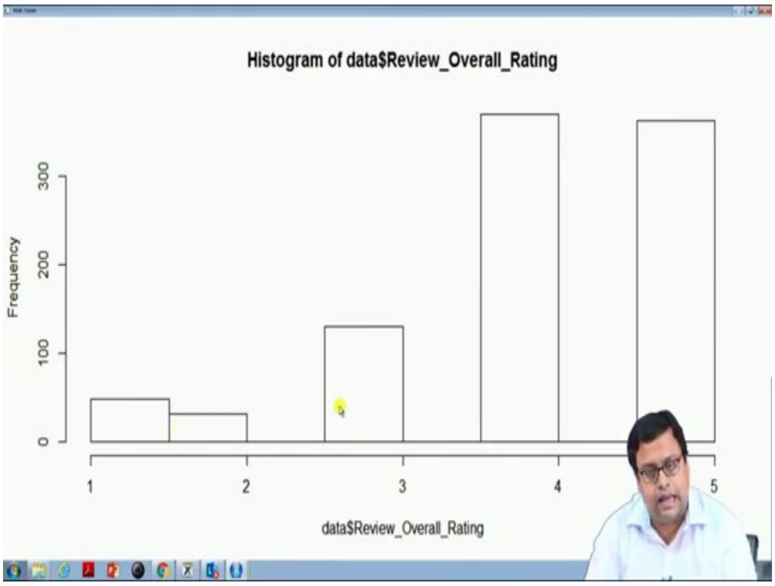


```
data <- read.csv("Hotel_Review.csv", header=TRUE)

str(data)

data$Review_Overall_Rating=ifelse(data$Review_Overall_Rating>med
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> median(data$Review_Overall_Rating)
[1] 4
> hist(data$Review_Overall_Rating)
>
```

Data

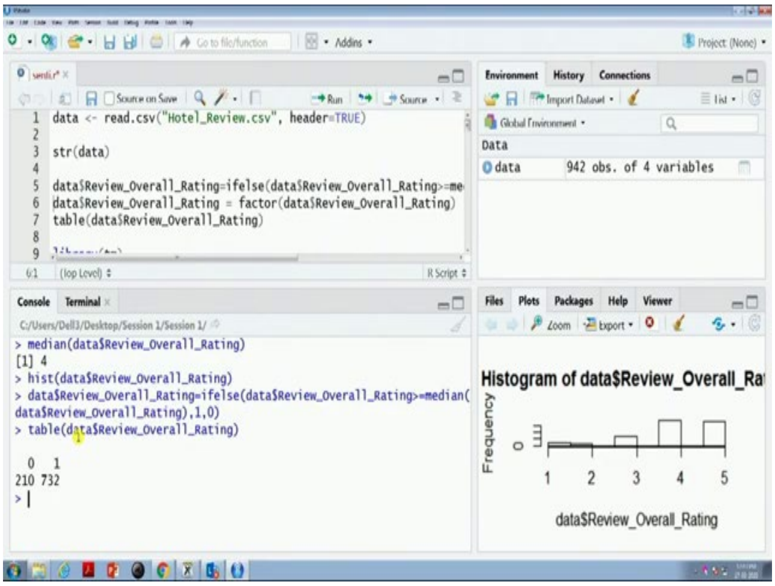
data	942 obs. of 4 variables



For example let us say if I just find out median of data dollar review overall rating. So that is coming up to be 4. So frankly speaking anything any review rating which is and if I just put a histogram of the same thing this is how it looks like the histogram. So which one will I say positive review and which one will I say negative review.

So can I say that all these 5 are positive, these 4 probably are also positive so majorly biased towards positive review, but 3, 2, 1 are obviously negative reviews these guys are all negative review so I can say that. So if 3, 2, 1 are negative reviews 4 and 5 are positive review.

(Refer Slide Time: 10:11)



```
data <- read.csv("Hotel_Review.csv", header=TRUE)

str(data)

data$Review_Overall_Rating=ifelse(data$Review_Overall_Rating>=me
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)
```

```
> median(data$Review_Overall_Rating)
[1] 4
> hist(data$Review_Overall_Rating)
> data$Review_Overall_Rating=ifelse(data$Review_Overall_Rating>=median(
data$Review_Overall_Rating),1,0)
> table(data$Review_Overall_Rating)

  0   1
210 732
> 
```

Data
data 942 obs. of 4 variables

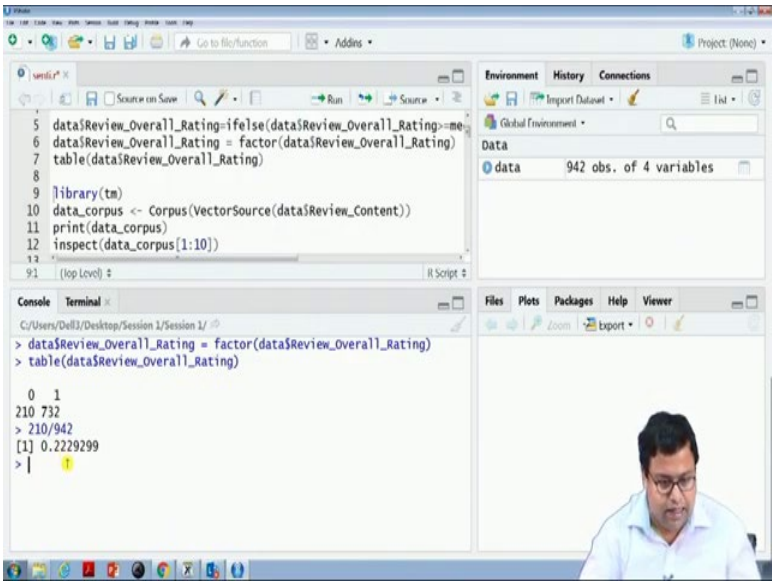
Histogram of data$Review_Overall_Ra
Frequency
0
1
2
3
4
5
data$Review_Overall_Rating

So what I will do here is data dollar review overall rating is let us say if they are greater than or let us say greater than equal to if I just write greater than that means it should has to be higher than 4.

If it is not higher than 4 then it will be negative. So data review overall rating is if else data dollar review overall rating greater than the median of that then it is 1 otherwise 0. So change this thing to 1, 0 is what I am trying to do. So I am changing it to 1, 0. So I will later rather say greater than equal to okay because if I just write greater than that means only 5 will be considered as positive review 1, 2, 3, 4 will be considered as negative review which is not the case probably.

So greater than equal to I will say up to 4 it is 4 and 5 it is positive review and otherwise it is negative review. So I am changing it to 1, 0. So now if I just try to see table data dollar review overall rating. I have 210 0s and 732 1s so this is my situation right now. So next what will I do I will make them factor.

(Refer Slide Time: 11:32)



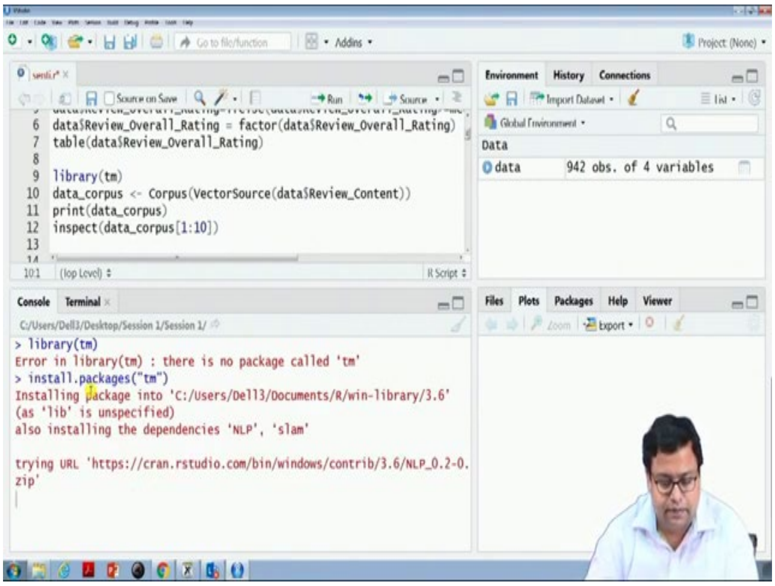
```
data$Review_Overall_Rating=ifelse(data$Review_Overall_Rating>=me
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])
```

```
> data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
> table(data$Review_Overall_Rating)

  0   1 
210 732 
> 210/942
[1] 0.2229299
> 
```

Data
data 942 obs. of 4 variables

This 1, 0 instead of considering as 0 and 1, two numbers we should consider them as two groups, two factors. So I am changing it to two factors. Now if I want to see the table of this thing the same thing whatever I told will be there so 210. If we get an idea 210 / 942 that comes up to be 22 % are negative and around 78 % are positive, but still this is biased but this not heavily biased so that we have to use something else it is not like that it is biased, but we can still use this one. So we will be going ahead.

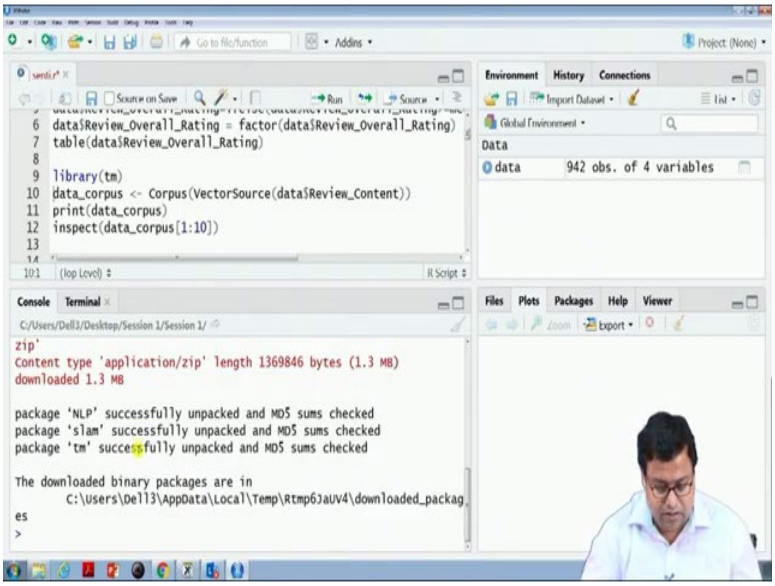
(Refer Slide Time: 12:18)



```
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])
```

```
> library(tm)
Error in library(tm) : there is no package called 'tm'
> install.packages("tm")
Installing package into 'C:/Users/Dell3/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
also installing the dependencies 'NLP', 'slam'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/NLP_0.2-0.
zip'
```

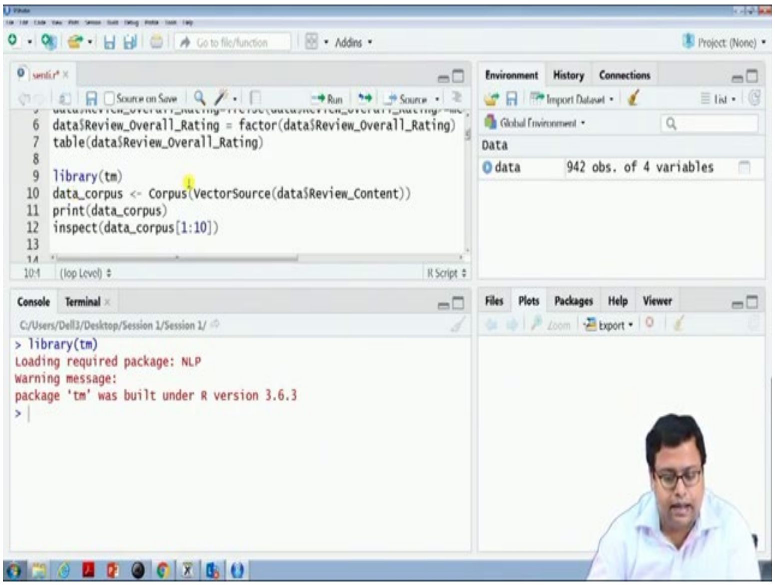
Data
data 942 obs. of 4 variables



```
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])
```

```
zip'
Content type 'application/zip' length 1369846 bytes (1.3 MB)
downloaded 1.3 MB

package 'NLP' successfully unpacked and MD5 sums checked
package 'slam' successfully unpacked and MD5 sums checked
package 'tm' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Dell3\AppData\Local\Temp\Rtmp6JaUV4\downloaded_packag
es
>
```

Data
data 942 obs. of 4 variables



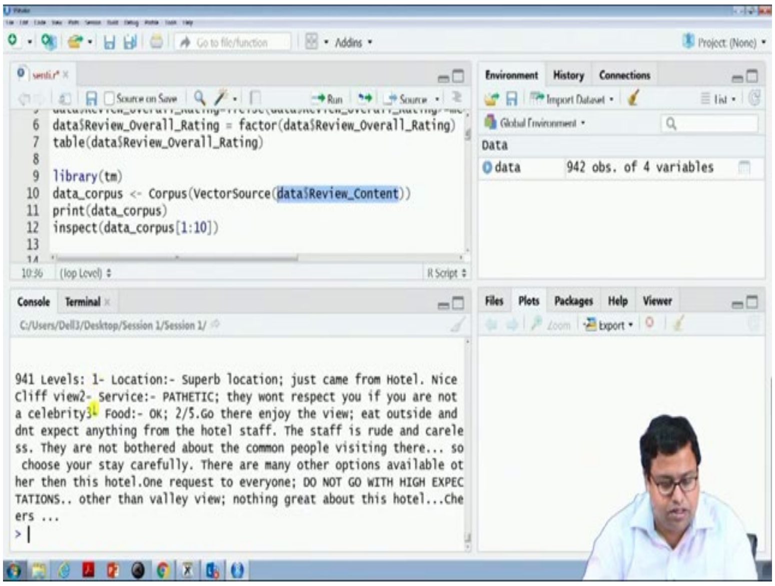
```
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])
```

```
> library(tm)
Loading required package: NLP
Warning message:
package 'tm' was built under R version 3.6.3
>
```

Data
data 942 obs. of 4 variables

Now the library that I call is this tm library text mining library. So I will call this library there is no package. So what we will do is we will install this thing, install tm we will take some time. Okay it has install now, so now I will call this library called tm. So I called this library it is saying that tm has been built under 3.6.3, but it is okay as long as the warning message and as long as the other functionalities are there we can go ahead.

(Refer Slide Time: 13:06)

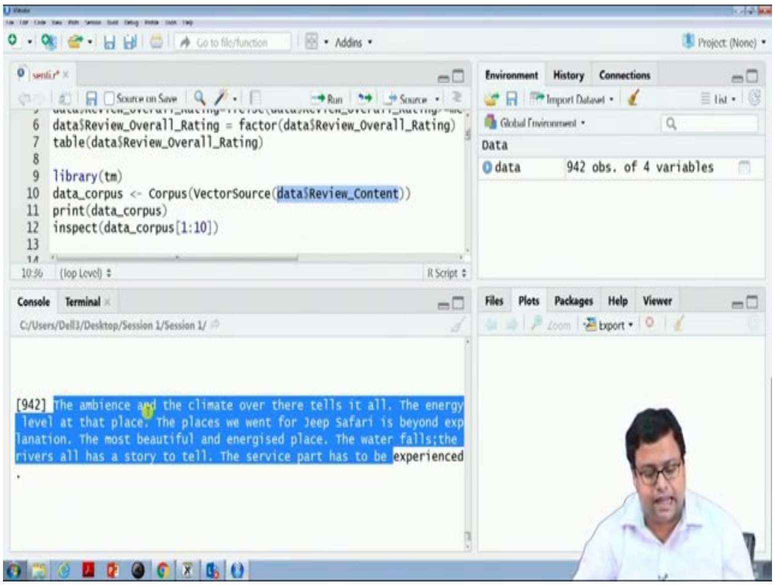


```r
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])
```

Data
data 942 obs. of 4 variables

C:/Users/Dell3/Desktop/Session 1/Session 1/

```
941 Levels: 1- Location:- Superb location; just came from Hotel. Nice
Cliff view2- Service:- PATHETIC; they wont respect you if you are not
a celebrity3- Food:- OK; 2/5.Go there enjoy the view; eat outside and
dnt expect anything from the hotel staff. The staff is rude and carele
ss. They are not bothered about the common people visiting there... so
 choose your stay carefully. There are many other options available ot
her then this hotel.One request to everyone; DO NOT GO WITH HIGH EXPEC
TATIONS.. other than valley view; nothing great about this hotel...Che
ers ...
> |
```



```r
data$Review_Overall_Rating = factor(data$Review_Overall_Rating)
table(data$Review_Overall_Rating)

library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])
```

Data
data 942 obs. of 4 variables

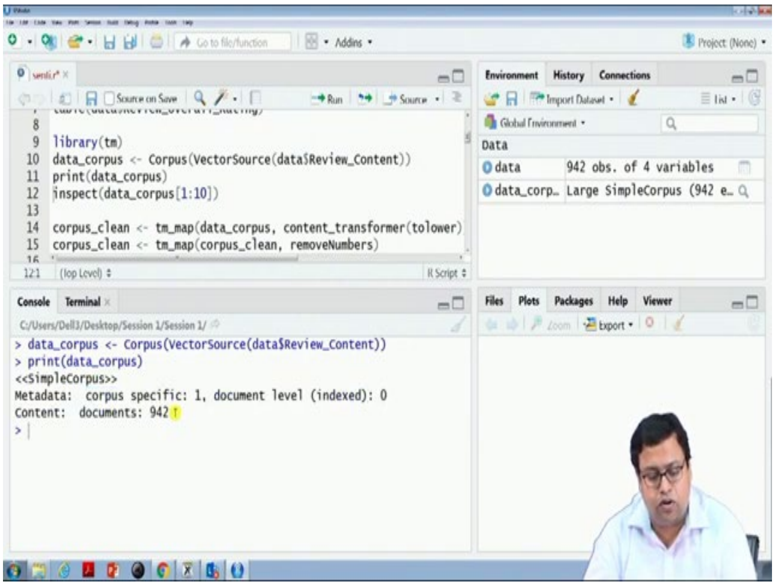
C:/Users/Dell3/Desktop/Session 1/Session 1/

```
[942] The ambience and the climate over there tells it all. The energy
 level at that place. The places we went for Jeep Safari is beyond exp
lanation. The most beautiful and energised place. The water falls;the
rivers all has a story to tell. The service part has to be experienced
.
```

So then as usual we are changing this data dollar review content which looks like this. So we are dealing with the review content. Why we are dealing with review content, somebody else will tell you sometimes later I will tell you that review content is something where the text is of significant size. On other than review I should not have actually click review content the whole thing because there are lots of content and it will get printed here, yeah so it got printed.

So this is something which is of huge text size. You will see that this is one content the ambience and the climate over there tells it all the energy blah, blah, blah so all of this the whole content is of significant size.

(Refer Slide Time: 13:51)

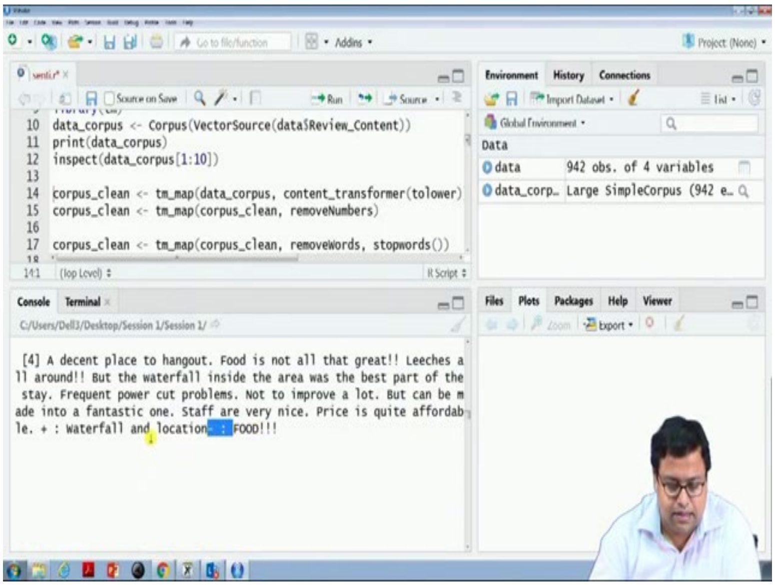


```
library(tm)
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])

corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
corpus_clean <- tm_map(corpus_clean, removeNumbers)
```

```
> data_corpus <- Corpus(VectorSource(data$Review_Content))
> print(data_corpus)
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 942
>
```

data   942 obs. of 4 variables

data_corp… Large SimpleCorpus (942 e…

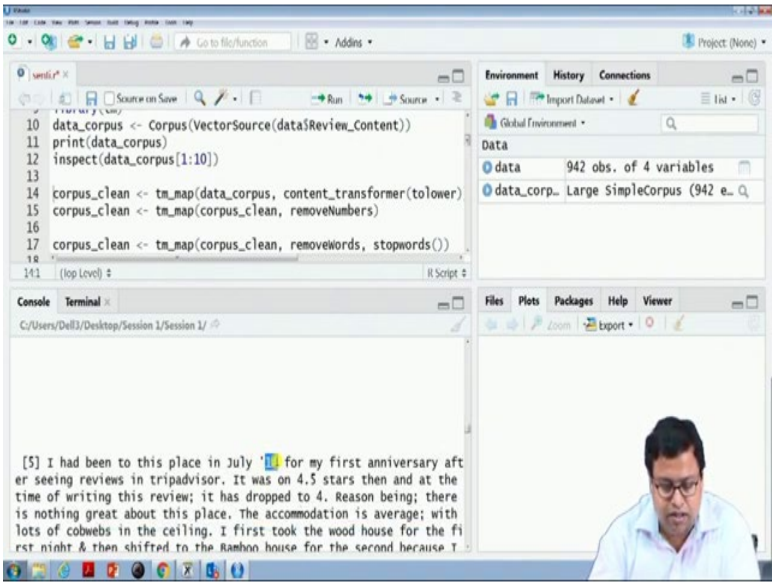


```
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])

corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
corpus_clean <- tm_map(corpus_clean, removeNumbers)

corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
```

```
 [4] A decent place to hangout. Food is not all that great!! Leeches a
ll around!! But the waterfall inside the area was the best part of the
 stay. Frequent power cut problems. Not to improve a lot. But can be m
ade into a fantastic one. Staff are very nice. Price is quite affordab
le. + : Waterfall and location - : FOOD!!!
```

data   942 obs. of 4 variables

data_corp… Large SimpleCorpus (942 e…



```r
data_corpus <- Corpus(VectorSource(data$Review_Content))
print(data_corpus)
inspect(data_corpus[1:10])

corpus_clean <- tm_map(data_corpus, content_transformer(tolower)
corpus_clean <- tm_map(corpus_clean, removeNumbers)

corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
```

```
Data
data          942 obs. of 4 variables
data_corp…    Large SimpleCorpus (942 e…
```

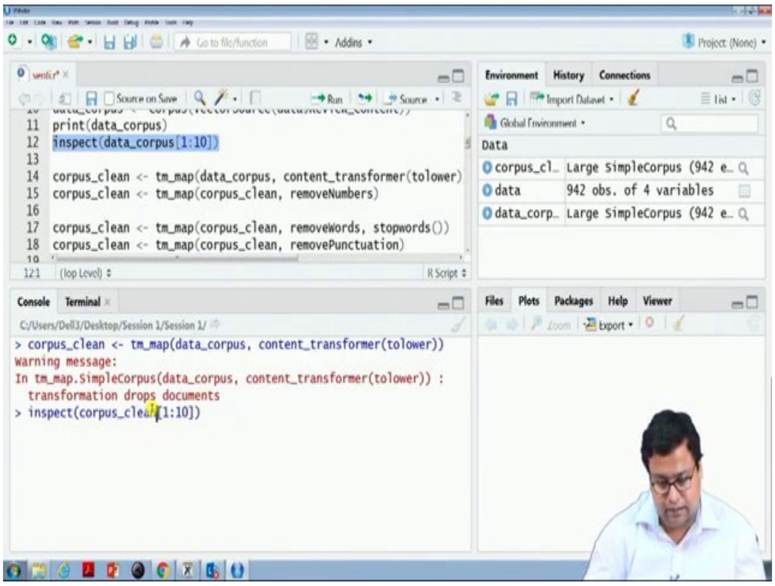
```
C:/Users/Dell3/Desktop/Session 1/Session 1/

 [5] I had been to this place in July '11 for my first anniversary aft
er seeing reviews in tripadvisor. It was on 4.5 stars then and at the
time of writing this review; it has dropped to 4. Reason being; there
is nothing great about this place. The accommodation is average; with
lots of cobwebs in the ceiling. I first took the wood house for the fi
rst night & then shifted to the Bamboo house for the second because I
```

On the other hand if I just printed review title review titles were small so there are not enough words to analyze the review title that whether they are impacting the overall rating, but they might be impacting more than other things.

So anyways we will deal with review content so this content we change it to corpus form as we did in the last video. Now once we have change it to corpus form we print the corpus and the corpus has basically a metadata which has 942 documents and if I want to inspect the first 10 documents these are the first 10 documents. So 1, 2 so these are all text that is there. Now you see in this text there are lots of punctuations marks are there.

Lot of numbers are there, lots of various other things are there. So these things may not have any meaning. So the first job as we did any dataset is to probably do a little of preprocessing of the data.

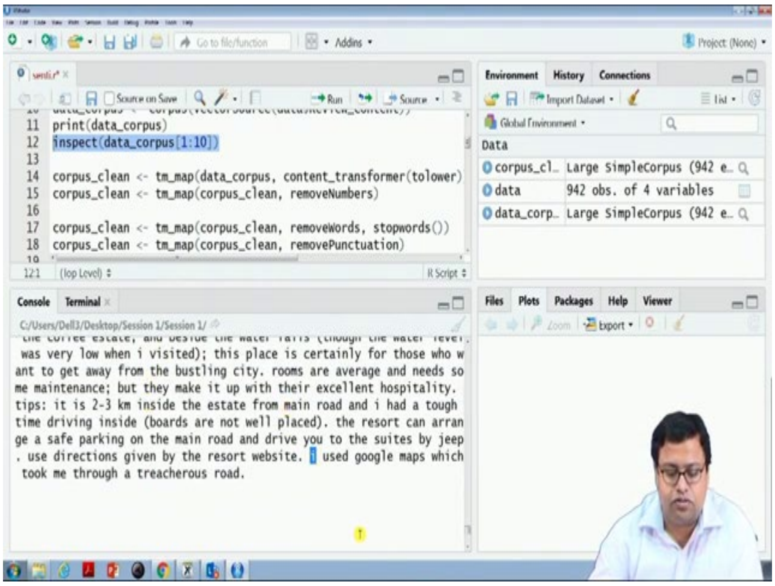
(Refer Slide Time: 14:55)



```
print(data_corpus)
inspect(data_corpus[1:10])

corpus_clean <- tm_map(data_corpus, content_transformer(tolower)
corpus_clean <- tm_map(corpus_clean, removeNumbers)

corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
corpus_clean <- tm_map(corpus_clean, removePunctuation)
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
Warning message:
In tm_map.SimpleCorpus(data_corpus, content_transformer(tolower)) :
  transformation drops documents
> inspect(corpus_clean[1:10])
```

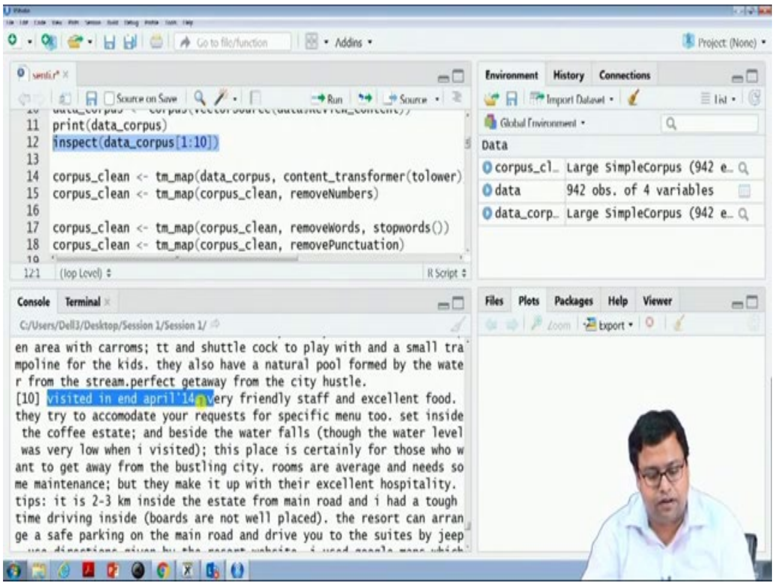
```
Data
corpus_cl…   Large SimpleCorpus (942 e…
data         942 obs. of 4 variables
data_corp…   Large SimpleCorpus (942 e…
```



```
print(data_corpus)
inspect(data_corpus[1:10])

corpus_clean <- tm_map(data_corpus, content_transformer(tolower)
corpus_clean <- tm_map(corpus_clean, removeNumbers)

corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
corpus_clean <- tm_map(corpus_clean, removePunctuation)
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
 the coffee estate, and beside the water falls (though the water level
 was very low when i visited); this place is certainly for those who w
ant to get away from the bustling city. rooms are average and needs so
me maintenance; but they make it up with their excellent hospitality.
tips: it is 2-3 km inside the estate from main road and i had a tough
time driving inside (boards are not well placed). the resort can arran
ge a safe parking on the main road and drive you to the suites by jeep
. use directions given by the resort website. i used google maps which
 took me through a treacherous road.
```

```
Data
corpus_cl…   Large SimpleCorpus (942 e…
data         942 obs. of 4 variables
data_corp…   Large SimpleCorpus (942 e…
```



```r
print(data_corpus)
inspect(data_corpus[1:10])

corpus_clean <- tm_map(data_corpus, content_transformer(tolower)
corpus_clean <- tm_map(corpus_clean, removeNumbers)

corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
corpus_clean <- tm_map(corpus_clean, removePunctuation)
```

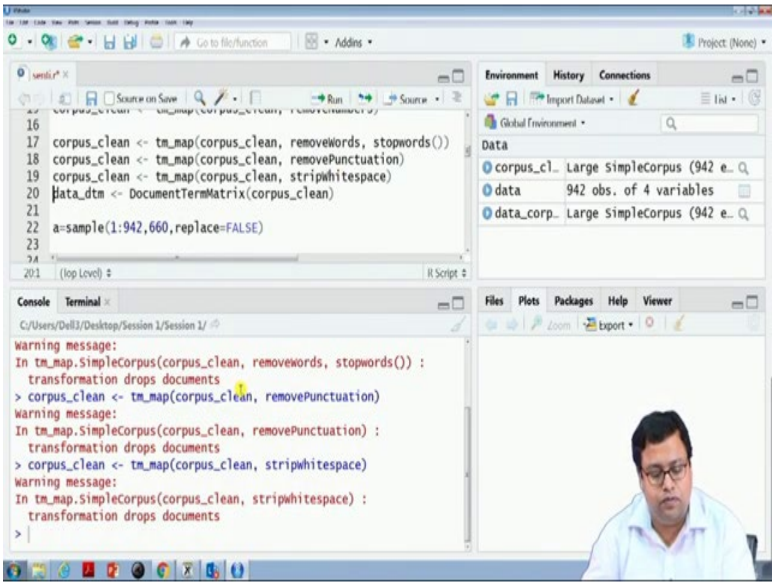
```
Data
corpus_cl…  Large SimpleCorpus (942 e…
data        942 obs. of 4 variables
data_corp…  Large SimpleCorpus (942 e…
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
en area with carroms; tt and shuttle cock to play with and a small tra
mpoline for the kids. they also have a natural pool formed by the wate
r from the stream.perfect getaway from the city hustle.
[10] visited in end april'14. very friendly staff and excellent food.
they try to accomodate your requests for specific menu too. set inside
 the coffee estate; and beside the water falls (though the water level
 was very low when i visited); this place is certainly for those who w
ant to get away from the bustling city. rooms are average and needs so
me maintenance; but they make it up with their excellent hospitality.
tips: it is 2-3 km inside the estate from main road and i had a tough
time driving inside (boards are not well placed). the resort can arran
ge a safe parking on the main road and drive you to the suites by jeep
```

So this is very important mode so in the text context that you have to preprocess your data. So what we will do to preprocess? I first content transform to lower that means I change the whole dataset to the lower case.

So this is equal to sign, this is basically equal to sign. We have discussed in the first class that this is also how equal to signs are written. So I run that and in corpus clean the all everything that is there in corpus clean is basically a in the lower form. So if I just write corpus clean you see that here this i has been changed to its lower case. Let us say visited in end April vary so everything after full stop has become lower case.

(Refer Slide Time: 16:00)



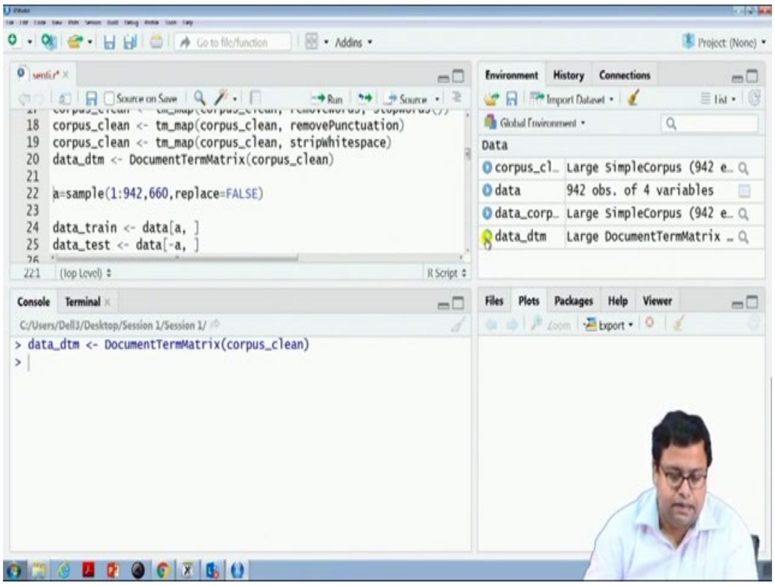
```r
corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
corpus_clean <- tm_map(corpus_clean, removePunctuation)
corpus_clean <- tm_map(corpus_clean, stripWhitespace)
data_dtm <- DocumentTermMatrix(corpus_clean)

a=sample(1:942,660,replace=FALSE)
```

Data	
corpus_cl…	Large SimpleCorpus (942 e…
data	942 obs. of 4 variables
data_corp…	Large SimpleCorpus (942 e…

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
Warning message:
In tm_map.SimpleCorpus(corpus_clean, removeWords, stopwords()) :
  transformation drops documents
> corpus_clean <- tm_map(corpus_clean, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(corpus_clean, removePunctuation) :
  transformation drops documents
> corpus_clean <- tm_map(corpus_clean, stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(corpus_clean, stripWhitespace) :
  transformation drops documents
> 
```

So the whole dataset is right now in lower case then I clean, I remove the numbers I have remove the numbers then I remove the stop words so what are the stop words. Stop words are generic words which do not have any contribution in the meaning of the dataset so we remove the stop word, we remove punctuations, we remove white space. So one by one I am removing lots of warning messages are coming up right now we will not bother later we will see.

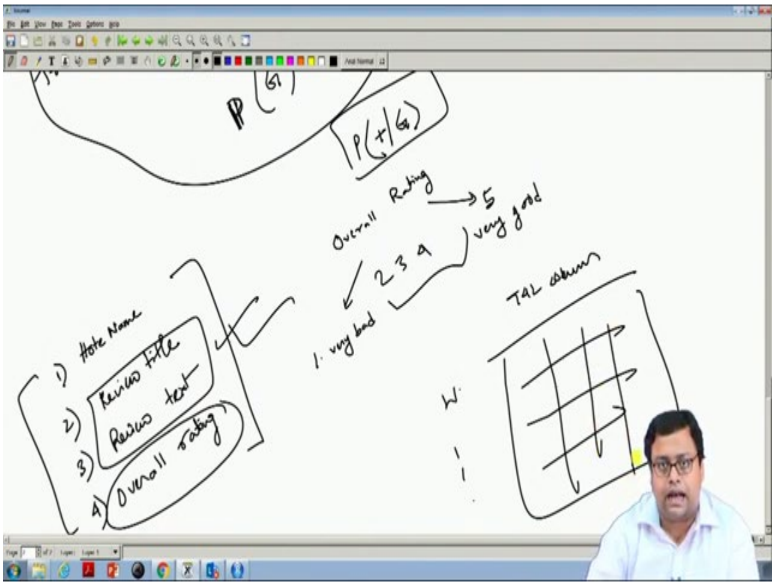
(Refer Slide Time: 16:31)



```
corpus_clean <- tm_map(corpus_clean, removePunctuation)
corpus_clean <- tm_map(corpus_clean, stripWhitespace)
data_dtm <- DocumentTermMatrix(corpus_clean)

a=sample(1:942,660,replace=FALSE)

data_train <- data[a, ]
data_test <- data[-a, ]
```

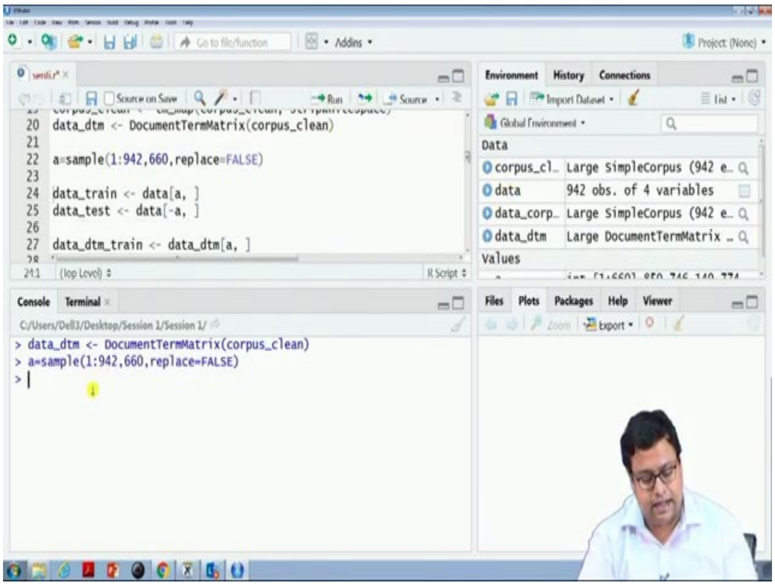
Data	
corpus_cl…	Large SimpleCorpus (942 e…
data	942 obs. of 4 variables
data_corp…	Large SimpleCorpus (942 e…
data_dtm	Large DocumentTermMatrix …

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> data_dtm <- DocumentTermMatrix(corpus_clean)
>
```



And then what I do is with this dataset I create a document term matrix. So in my document term matrix it is also a large document and basically it is document of matrix looks like this that there are lots of words, word 1, 2 and there are 942 columns and this is basically a matrix where the counts are there, all the counters how many times one particular word is occurring in which particular review that is there.

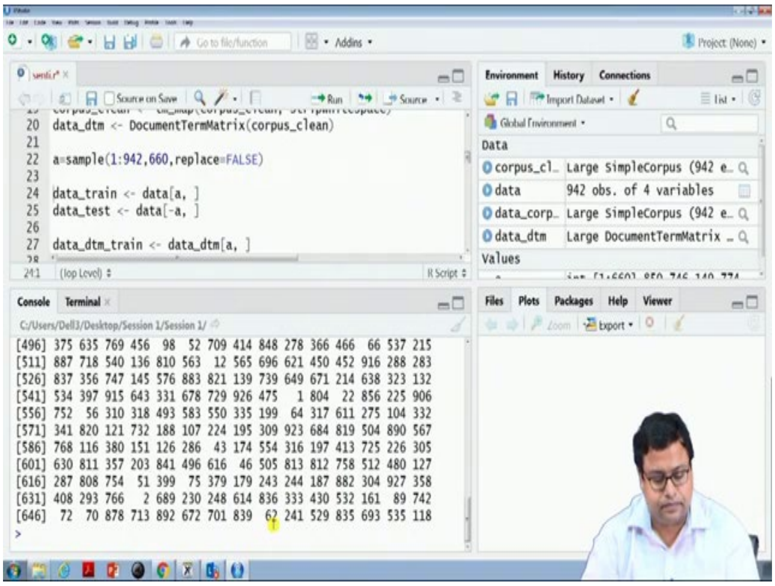
(Refer Slide Time: 17:05)



```
data_dtm <- DocumentTermMatrix(corpus_clean)

a=sample(1:942,660,replace=FALSE)

data_train <- data[a, ]
data_test <- data[-a, ]

data_dtm_train <- data_dtm[a, ]
```

Data	
corpus_cl…	Large SimpleCorpus (942 e…
data	942 obs. of 4 variables
data_corp…	Large SimpleCorpus (942 e…
data_dtm	Large DocumentTermMatrix …

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> data_dtm <- DocumentTermMatrix(corpus_clean)
> a=sample(1:942,660,replace=FALSE)
> |
```



```
data_dtm <- DocumentTermMatrix(corpus_clean)

a=sample(1:942,660,replace=FALSE)

data_train <- data[a, ]
data_test <- data[-a, ]

data_dtm_train <- data_dtm[a, ]
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
[496] 375 635 769 456  98  52 709 414 848 278 366 466  66 537 215
[511] 887 718 540 136 810 563  12 565 696 621 450 452 916 288 283
[526] 837 356 747 145 576 883 821 139 739 649 671 214 638 323 132
[541] 534 397 915 643 331 678 729 926 475   1 804  22 856 225 906
[556] 752  56 310 318 493 583 550 335 199  64 317 611 275 104 332
[571] 341 820 121 732 188 107 224 195 309 923 684 819 504 890 567
[586] 768 116 380 151 126 286  43 174 554 316 197 413 725 226 305
[601] 630 811 357 203 841 496 616  46 505 813 812 758 512 480 127
[616] 287 808 754  51 399  75 379 179 243 244 187 882 304 927 358
[631] 408 293 766   2 689 230 248 614 836 333 430 532 161  89 742
[646]  72  70 878 713 892 672 701 839  62 241 529 835 693 535 118
>
```

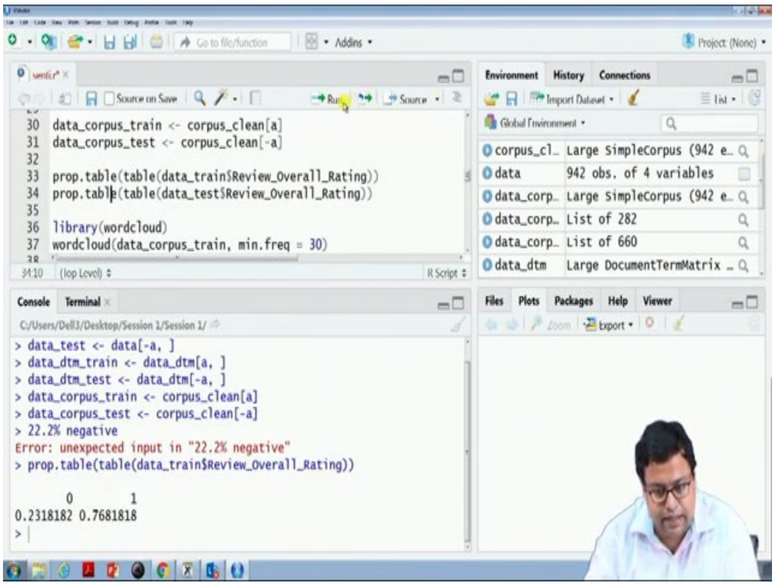
Now obviously I have to break it in training and testing. So if you remember in the last class we have done training testing by taking the first 70 % or 80 % as training and the last 20 % as testing that we cannot do here because there are various people. So your past review is actually sometimes impact your future review. So how the person who has reviewed last, the last review that I see.

When I go and write my review is also see the last review and if I see that review is positive sometimes my mind also gets diverted towards positivity or negativity. So all I am trying to say is that other people’s review might impact you. So that is why we cannot take the top 400 as testing and lower 200 as testing or training something like it we have to do it randomly. So for that I am using a sample function.

So sample function samples 660 observations from 1 to 942 in this column I create vector called 1 to 942 out of that 660 observations will be randomly picked up and replacement is equal to false means once it is picked up it will never be picked up again. It is not a sampling with replacing, it is sampling without replacing means if 66 has been selected once 66 this particular number cannot be selected once more.

So that kind of sampling we are doing and putting it in a. So my a is right now some random numbers that got generated from 1 to 942 and the length is 660.

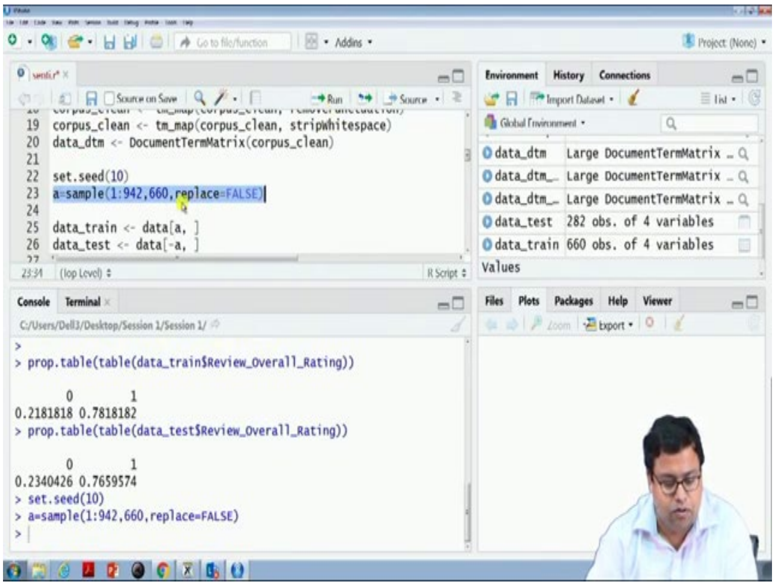
(Refer Slide Time: 18:51)



```
data_corpus_train <- corpus_clean[a]
data_corpus_test <- corpus_clean[-a]

prop.table(table(data_train$Review_Overall_Rating))
prop.table(table(data_test$Review_Overall_Rating))

library(wordcloud)
wordcloud(data_corpus_train, min.freq = 30)
```

corpus_cl_	Large SimpleCorpus (942 e_
data	942 obs. of 4 variables
data_corp_	Large SimpleCorpus (942 e_
data_corp_	List of 282
data_corp_	List of 660
data_dtm	Large DocumentTermMatrix _

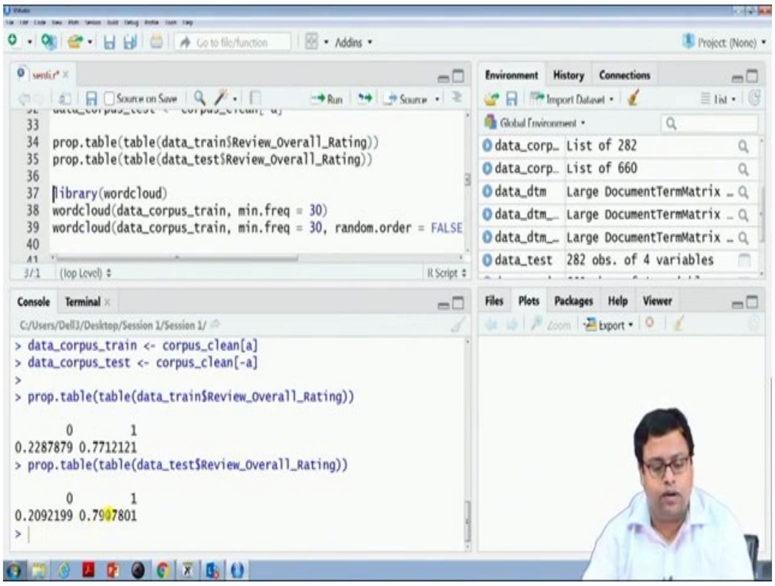
```
> data_test <- data[-a, ]
> data_dtm_train <- data_dtm[a, ]
> data_dtm_test <- data_dtm[-a, ]
> data_corpus_train <- corpus_clean[a]
> data_corpus_test <- corpus_clean[-a]
> 22.2% negative
Error: unexpected input in "22.2% negative"
> prop.table(table(data_train$Review_Overall_Rating))

        0         1 
0.2318182 0.7681818 
> 
```



```
corpus_clean <- tm_map(corpus_clean, stripWhitespace)
data_dtm <- DocumentTermMatrix(corpus_clean)

set.seed(10)
a=sample(1:942,660,replace=FALSE)

data_train <- data[a, ]
data_test <- data[-a, ]
```

data_dtm	Large DocumentTermMatrix _
data_dtm_	Large DocumentTermMatrix _
data_dtm_	Large DocumentTermMatrix _
data_test	282 obs. of 4 variables
data_train	660 obs. of 4 variables
Values	

```
> 
> prop.table(table(data_train$Review_Overall_Rating))

        0         1 
0.2181818 0.7818182 
> prop.table(table(data_test$Review_Overall_Rating))

        0         1 
0.2340426 0.7659574 
> set.seed(10)
> a=sample(1:942,660,replace=FALSE)
> 
```



```
prop.table(table(data_train$Review_Overall_Rating))
prop.table(table(data_test$Review_Overall_Rating))

library(wordcloud)
wordcloud(data_corpus_train, min.freq = 30)
wordcloud(data_corpus_train, min.freq = 30, random.order = FALSE
```

```
data_corp… List of 282
data_corp… List of 660
data_dtm   Large DocumentTermMatrix …
data_dtm_… Large DocumentTermMatrix …
data_dtm_… Large DocumentTermMatrix …
data_test  282 obs. of 4 variables
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> data_corpus_train <- corpus_clean[a]
> data_corpus_test <- corpus_clean[-a]
>
> prop.table(table(data_train$Review_Overall_Rating))

        0         1
0.2287879 0.7712121
> prop.table(table(data_test$Review_Overall_Rating))

        0         1
0.2092199 0.7907801
>
```

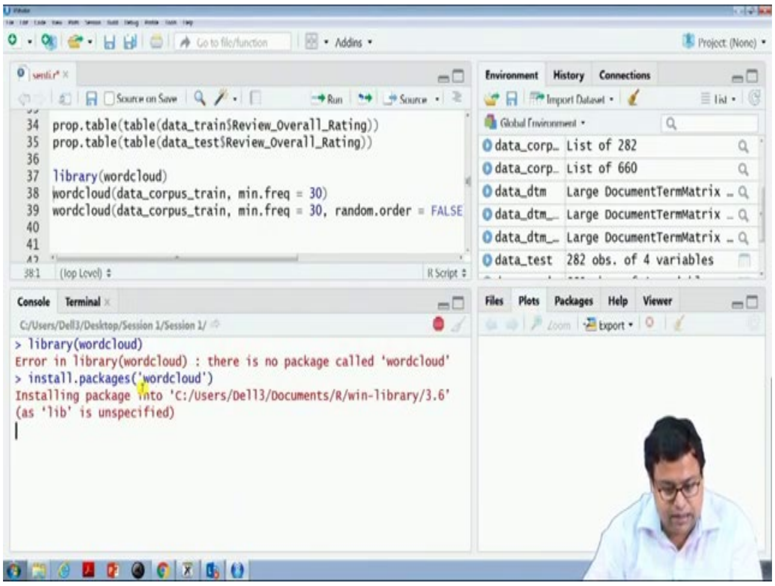
Now with this row names I am creating the training data and -a means the rest of the row names goes to the testing data so data dollar train. Similarly, if you remember in the last class also we created training, testing for the DTM and training testing for the clean corpus.

Because we will create I would say certain kind of word clouds using the corpus and to see that whether word is a way to find out whether that it is a positive review or negative review and then we will be doing our Naive Bayes using the document term matrix. So next what I do is as I told that there are if you remember 22.2 % positive, sorry negative this was there 22.2 % negative.

So if I just see currently what is the case you will see that 23.1 % are negative in your training data and your testing data 20.2 %. So not very different it is little bit different not very different. So I can probably sample once more and then I can do it to check that whether the prop table has improved a little bit or not. So let me just try out that so 21 and 23 it is okay I think that is fine.

So to have a replaceable or something which is obvious whatever I get and you get will be same you write set seed set .seed is let us say some number any number 10. So now my results and your results will come same. So now set seed and this once more and then the training data, testing data once more then the prop tables and etcetera. So the document term matrix and proportion table. So 22.77 very good and here also 21 and 0.79 so we will go ahead with this.

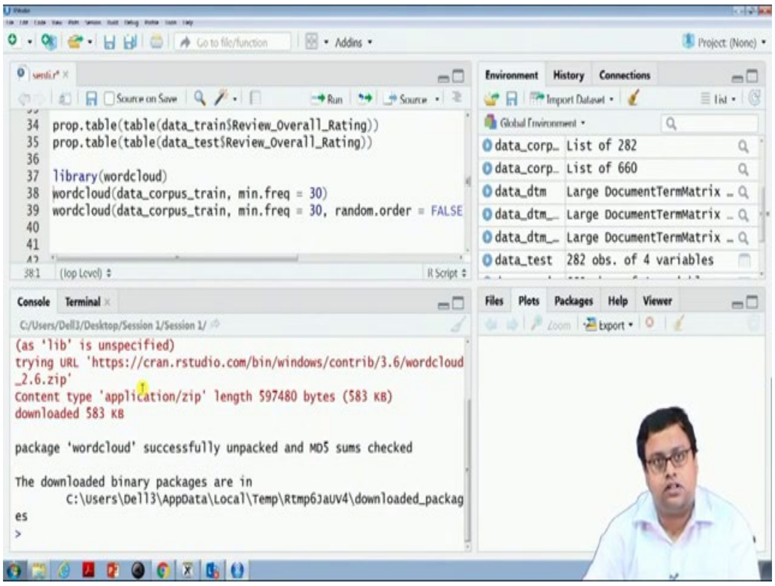
(Refer Slide Time: 21:00)



```
prop.table(table(data_train$Review_Overall_Rating))
prop.table(table(data_test$Review_Overall_Rating))

library(wordcloud)
wordcloud(data_corpus_train, min.freq = 30)
wordcloud(data_corpus_train, min.freq = 30, random.order = FALSE
```

```
data_corp.. List of 282
data_corp.. List of 660
data_dtm    Large DocumentTermMatrix ..
data_dtm_.. Large DocumentTermMatrix ..
data_dtm_.. Large DocumentTermMatrix ..
data_test   282 obs. of 4 variables
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> library(wordcloud)
Error in library(wordcloud) : there is no package called 'wordcloud'
> install.packages('wordcloud')
Installing package into 'C:/Users/Dell3/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
```



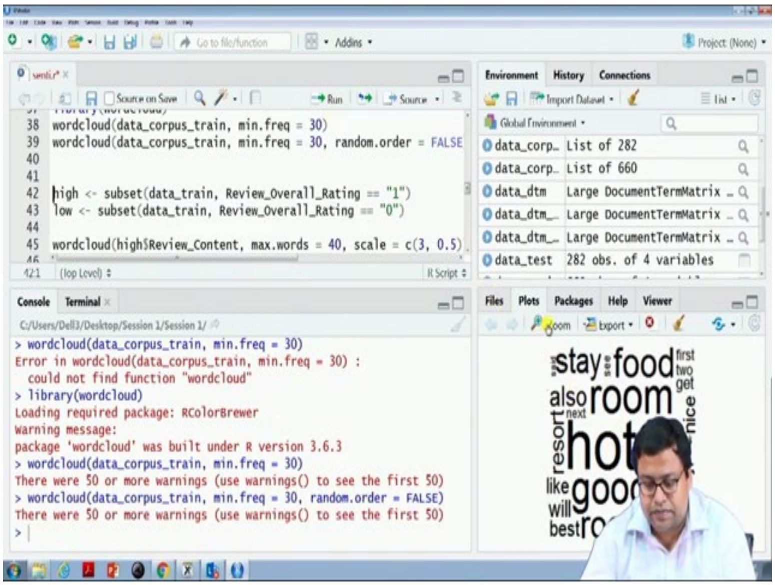
```
prop.table(table(data_train$Review_Overall_Rating))
prop.table(table(data_test$Review_Overall_Rating))

library(wordcloud)
wordcloud(data_corpus_train, min.freq = 30)
wordcloud(data_corpus_train, min.freq = 30, random.order = FALSE
```

```
data_corp.. List of 282
data_corp.. List of 660
data_dtm    Large DocumentTermMatrix ..
data_dtm_.. Large DocumentTermMatrix ..
data_dtm_.. Large DocumentTermMatrix ..
data_test   282 obs. of 4 variables
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/wordcloud
_2.6.zip'
Content type 'application/zip' length 597480 bytes (583 KB)
downloaded 583 KB

package 'wordcloud' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Dell3\AppData\Local\Temp\Rtmp6JaUV4\downloaded_packag
es
>
```

Now what I do is I call a library called word cloud. So let us see okay so this package not there, so install . packages word cloud. So let us install this one, so it got installed. Now with this package I run this thing.

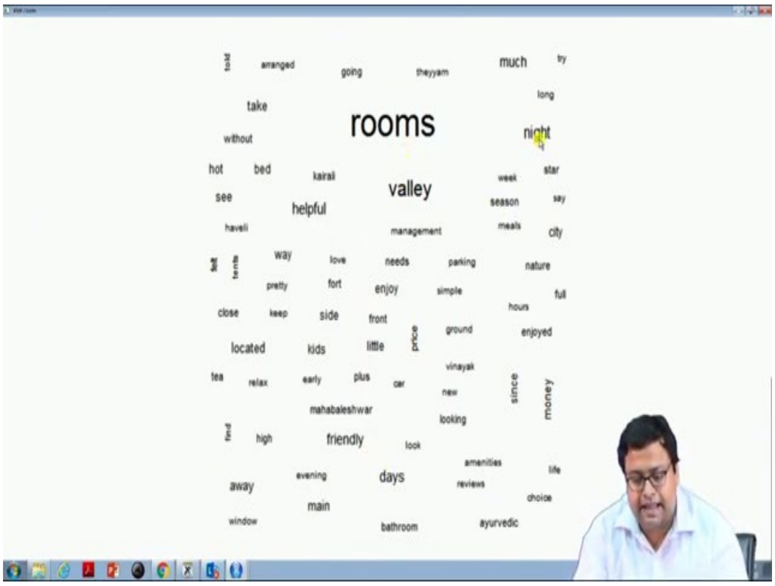
(Refer Slide Time: 21:22)

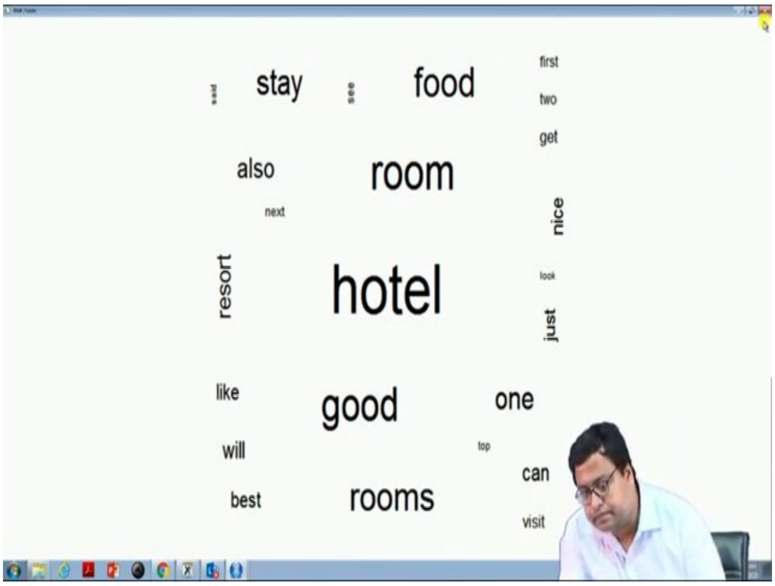


```
wordcloud(data_corpus_train, min.freq = 30)
wordcloud(data_corpus_train, min.freq = 30, random.order = FALSE


high <- subset(data_train, Review_Overall_Rating == "1")
low <- subset(data_train, Review_Overall_Rating == "0")

wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5)
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> wordcloud(data_corpus_train, min.freq = 30)
Error in wordcloud(data_corpus_train, min.freq = 30) :
  could not find function "wordcloud"
> library(wordcloud)
Loading required package: RColorBrewer
Warning message:
package 'wordcloud' was built under R version 3.6.3
> wordcloud(data_corpus_train, min.freq = 30)
There were 50 or more warnings (use warnings() to see the first 50)
> wordcloud(data_corpus_train, min.freq = 30, random.order = FALSE)
There were 50 or more warnings (use warnings() to see the first 50)
> |
```

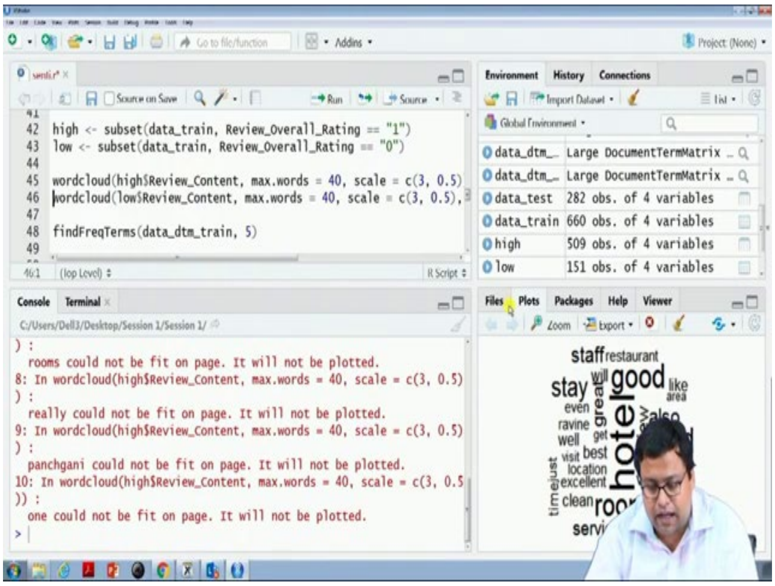
```
data_corp..   List of 282
data_corp..   List of 660
data_dtm      Large DocumentTermMatrix ..
data_dtm_..   Large DocumentTermMatrix ..
data_dtm_..   Large DocumentTermMatrix ..
data_test     282 obs. of 4 variables
```





Okay, I have to call the package first and with this package I run this thing. So the most common word is basically rooms fair enough. And then valley, friendly, helpful, some night some other words are also there which is coming up. Now rooms has to be the most common word so I should have probably remove that as my stop words, but it is okay. Now if I do the same thing for the training data without this randomness then hotel, room etcetera comes in the middle. So hotel, room, good food, room and rooms are same. So I should have done something called stemming to remove this room and rooms thing and then the rest.

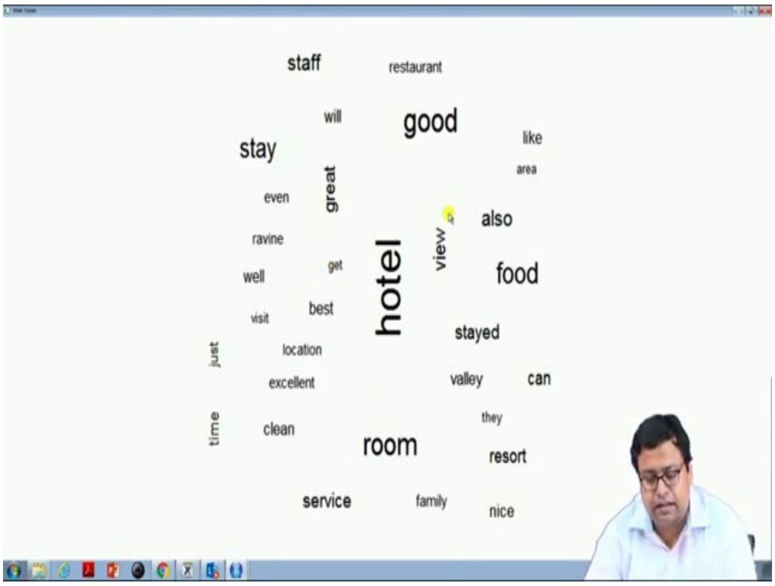
(Refer Slide Time: 22:27)



```
high <- subset(data_train, Review_Overall_Rating == "1")
low <- subset(data_train, Review_Overall_Rating == "0")

wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5)
wordcloud(low$Review_Content, max.words = 40, scale = c(3, 0.5),

findFreqTerms(data_dtm_train, 5)
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
) :
  rooms could not be fit on page. It will not be plotted.
8: In wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5)
) :
  really could not be fit on page. It will not be plotted.
9: In wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5)
) :
  panchgani could not be fit on page. It will not be plotted.
10: In wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5
)) :
  one could not be fit on page. It will not be plotted.
> 
```

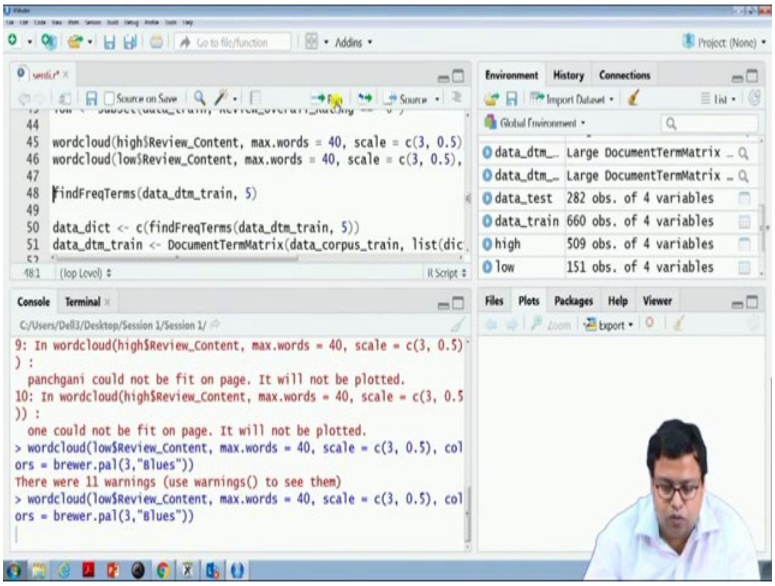
```
data_dtm_…   Large DocumentTermMatrix …
data_dtm_…   Large DocumentTermMatrix …
data_test    282 obs. of 4 variables
data_train   660 obs. of 4 variables
high         509 obs. of 4 variables
low          151 obs. of 4 variables
```



Now I break the dataset based on high and low. So this is high rating while the overall rating is 1 and the low rating of training data while the overall rating is 0. So I am breaking the training dataset into further two halves high and low. Then I am creating a word cloud for high and I will create a word cloud for low.

And I will compare that whether they are coming similar or not so let us create a word cloud for high. So high in this case hotel is common, but good room, great these are the words that are coming up just check remember this. So good food, staff, stay, service, great these are the words that are coming up.

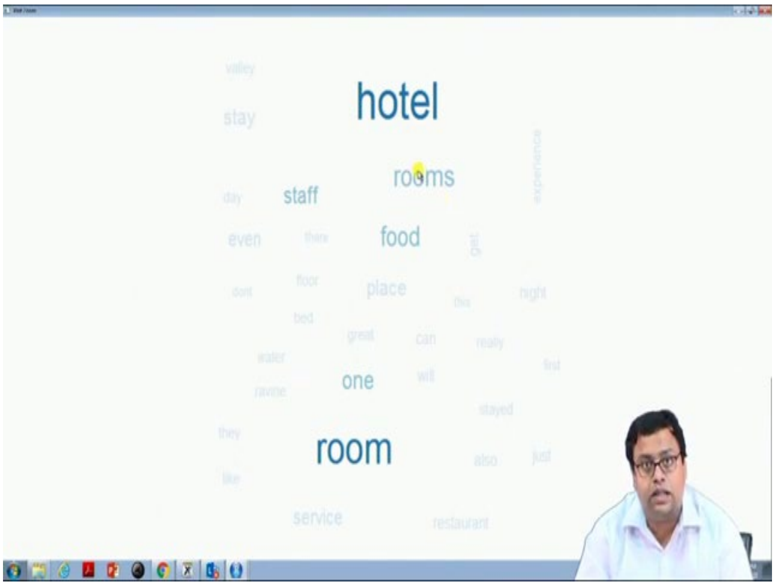
(Refer Slide Time: 23:20)



```
wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5)
wordcloud(low$Review_Content, max.words = 40, scale = c(3, 0.5),

findFreqTerms(data_dtm_train, 5)

data_dict <- c(findFreqTerms(data_dtm_train, 5))
data_dtm_train <- DocumentTermMatrix(data_corpus_train, list(dic
```

```
9: In wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5)
) :
  panchgani could not be fit on page. It will not be plotted.
10: In wordcloud(high$Review_Content, max.words = 40, scale = c(3, 0.5
)) :
  one could not be fit on page. It will not be plotted.
> wordcloud(low$Review_Content, max.words = 40, scale = c(3, 0.5), col
ors = brewer.pal(3,"Blues"))
There were 11 warnings (use warnings() to see them)
> wordcloud(low$Review_Content, max.words = 40, scale = c(3, 0.5), col
ors = brewer.pal(3,"Blues"))
```

data_dtm_	Large DocumentTermMatrix
data_dtm_	Large DocumentTermMatrix
data_test	282 obs. of 4 variables
data_train	660 obs. of 4 variables
high	509 obs. of 4 variables
low	151 obs. of 4 variables

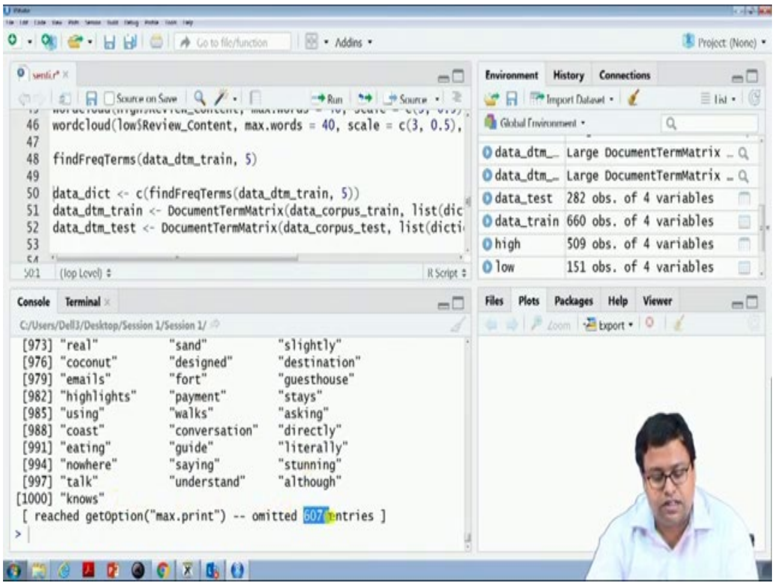


Now if I just try out the same thing with a different I will just clean this and then I will run with a different which is low then what comes up. So hotel room, rooms are common in both the cases, but see this good, great has gone away fair enough staff, food, room, rooms hotel these are all, but great is there in the much smaller size and practically good is not there if you see carefully.

So then this adjectives not the nouns, but probably the adjectives are something which can contribute towards your positiveness or negative and that makes sense that when we do a review of anything.

If you say that okay I am a good teacher, teacher is a noun, good is an adjective. So by this good or bad or lazy or very energetic these are the words that we use to define how this teacher is. So teacher is not creating any kind of positive or negative sentiment emotion it is not creating this adjectives are creating so that can be one learning from basic explanatory analysis of putting a word to.

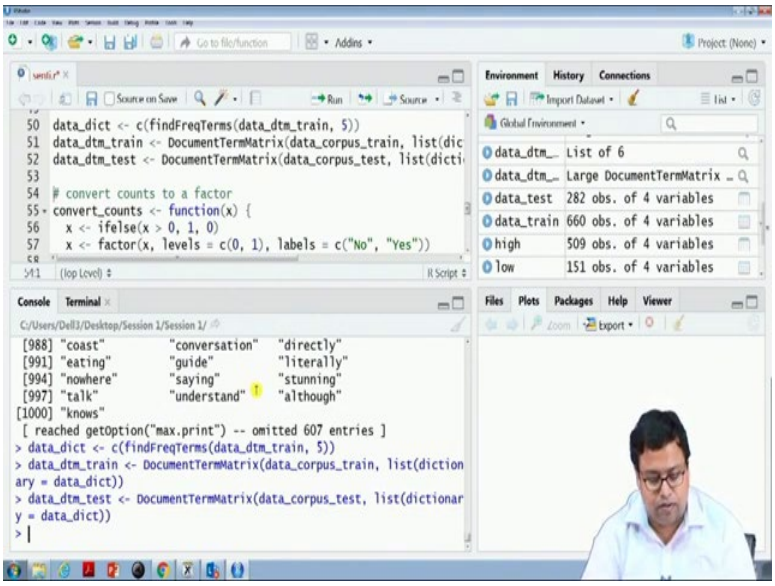
(Refer Slide Time: 24:36)



```
wordcloud(low$Review_Content, max.words = 40, scale = c(3, 0.5),

findFreqTerms(data_dtm_train, 5)

data_dict <- c(findFreqTerms(data_dtm_train, 5))
data_dtm_train <- DocumentTermMatrix(data_corpus_train, list(dic
data_dtm_test <- DocumentTermMatrix(data_corpus_test, list(dicti
```

data_dtm_	Large DocumentTermMatrix	
data_dtm_	Large DocumentTermMatrix	
data_test	282 obs. of 4 variables	
data_train	660 obs. of 4 variables	
high	509 obs. of 4 variables	
low	151 obs. of 4 variables	

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
 [973] "real"          "sand"          "slightly"
 [976] "coconut"       "designed"      "destination"
 [979] "emails"        "fort"          "guesthouse"
 [982] "highlights"    "payment"       "stays"
 [985] "using"         "walks"         "asking"
 [988] "coast"         "conversation"  "directly"
 [991] "eating"        "guide"         "literally"
 [994] "nowhere"       "saying"        "stunning"
 [997] "talk"          "understand"    "although"
[1000] "knows"
 [ reached getOption("max.print") -- omitted 607 entries ]
>
```



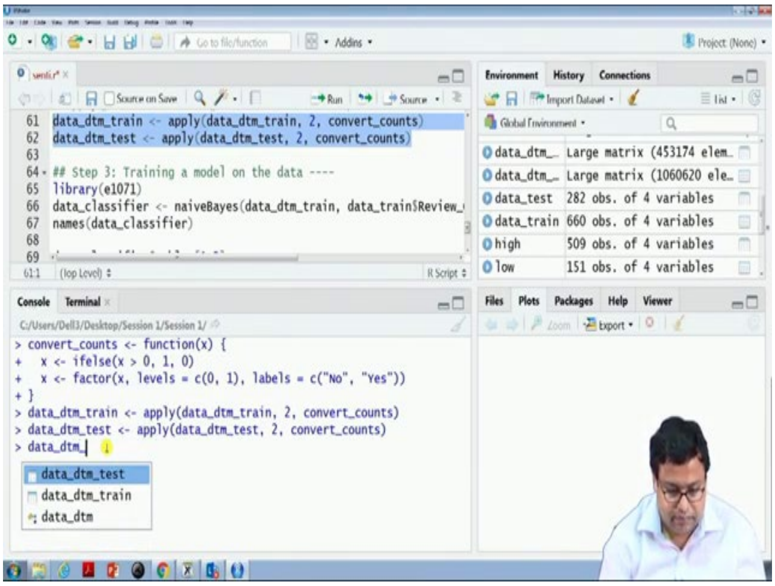
```
data_dict <- c(findFreqTerms(data_dtm_train, 5))
data_dtm_train <- DocumentTermMatrix(data_corpus_train, list(dic
data_dtm_test <- DocumentTermMatrix(data_corpus_test, list(dicti

# convert counts to a factor
convert_counts <- function(x) {
  x <- ifelse(x > 0, 1, 0)
  x <- factor(x, levels = c(0, 1), labels = c("No", "Yes"))
```

data_dtm_	List of 6	
data_dtm_	Large DocumentTermMatrix	
data_test	282 obs. of 4 variables	
data_train	660 obs. of 4 variables	
high	509 obs. of 4 variables	
low	151 obs. of 4 variables	

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
 [988] "coast"         "conversation"  "directly"
 [991] "eating"        "guide"         "literally"
 [994] "nowhere"       "saying"        "stunning"
 [997] "talk"          "understand"    "although"
[1000] "knows"
 [ reached getOption("max.print") -- omitted 607 entries ]
> data_dict <- c(findFreqTerms(data_dtm_train, 5))
> data_dtm_train <- DocumentTermMatrix(data_corpus_train, list(diction
ary = data_dict))
> data_dtm_test <- DocumentTermMatrix(data_corpus_test, list(dictionar
y = data_dict))
> |
```

Now what I will do I will find out the frequent terms up to 5 frequent terms. So there are 1607 entries which has occurred at least 5 times make a dictionary with this words I am making a dictionary with these words and then what I will do is I will create a document term matrix only with this word for the training data also and for the testing data also. So two document term matrix once more for the training data and for the testing data, but only with this 1607 words.

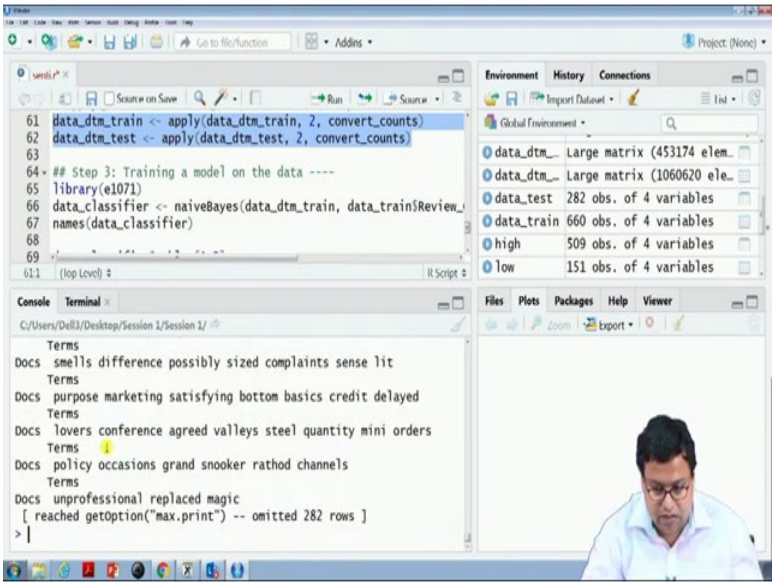
(Refer Slide Time: 25:10)



```
data_dtm_train <- apply(data_dtm_train, 2, convert_counts)
data_dtm_test <- apply(data_dtm_test, 2, convert_counts)

## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)
```

```
data_dtm_   Large matrix (453174 elem_
data_dtm_   Large matrix (1060620 ele_
data_test   282 obs. of 4 variables
data_train  660 obs. of 4 variables
high        509 obs. of 4 variables
low         151 obs. of 4 variables
```

```
> convert_counts <- function(x) {
+   x <- ifelse(x > 0, 1, 0)
+   x <- factor(x, levels = c(0, 1), labels = c("No", "Yes"))
+ }
> data_dtm_train <- apply(data_dtm_train, 2, convert_counts)
> data_dtm_test <- apply(data_dtm_test, 2, convert_counts)
> data_dtm_
   data_dtm_test
   data_dtm_train
   data_dtm
```

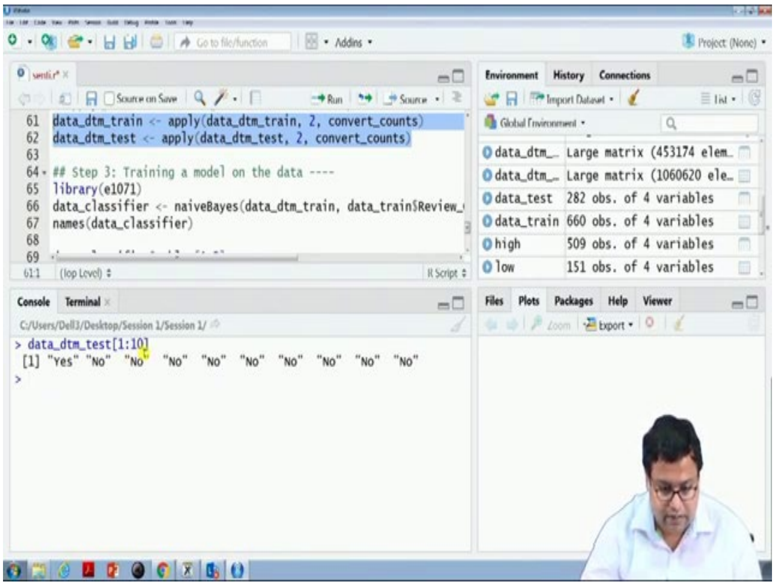


```
data_dtm_train <- apply(data_dtm_train, 2, convert_counts)
data_dtm_test <- apply(data_dtm_test, 2, convert_counts)

## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)
```

```
      Terms
Docs  smells difference possibly sized complaints sense lit
      Terms
Docs  purpose marketing satisfying bottom basics credit delayed
      Terms
Docs  lovers conference agreed valleys steel quantity mini orders
      Terms
Docs  policy occasions grand snooker rathod channels
      Terms
Docs  unprofessional replaced magic
 [ reached getOption("max.print") -- omitted 282 rows ]
>
```

Now remember document term matrix gives you counts I need 10 in the last week we have done that so I convert it to basically counts rather than from counts to 1, 0 yes no I am converting it to that. So it has done that. Now if I just check what is data DTM _ test if I just try to see that okay so this I will not be able to see this here.

(Refer Slide Time: 25:41)



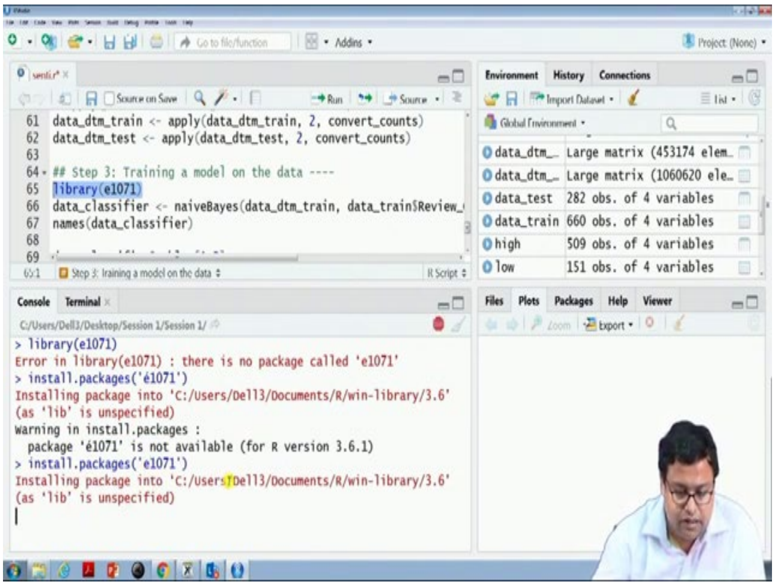
```
data_dtm_train <- apply(data_dtm_train, 2, convert_counts)
data_dtm_test <- apply(data_dtm_test, 2, convert_counts)

## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)
```

```
data_dtm_   Large matrix (453174 elem…
data_dtm_   Large matrix (1060620 ele…
data_test   282 obs. of 4 variables
data_train  660 obs. of 4 variables
high        509 obs. of 4 variables
low         151 obs. of 4 variables
```

```
C:/Users/Dell3/Desktop/Session 1/Session 1/
> data_dtm_test[1:10]
 [1] "Yes" "No"  "No"  "No"  "No"  "No"  "No"  "No"  "No"  "No"
>
```

So let us say data 10 1 to 10. Now this is not how I will see basically the data this is my training data. So these are one to the word characters and their corresponding these are the terms and these are basically the corresponding values 1, 2, 3, 4, 5.

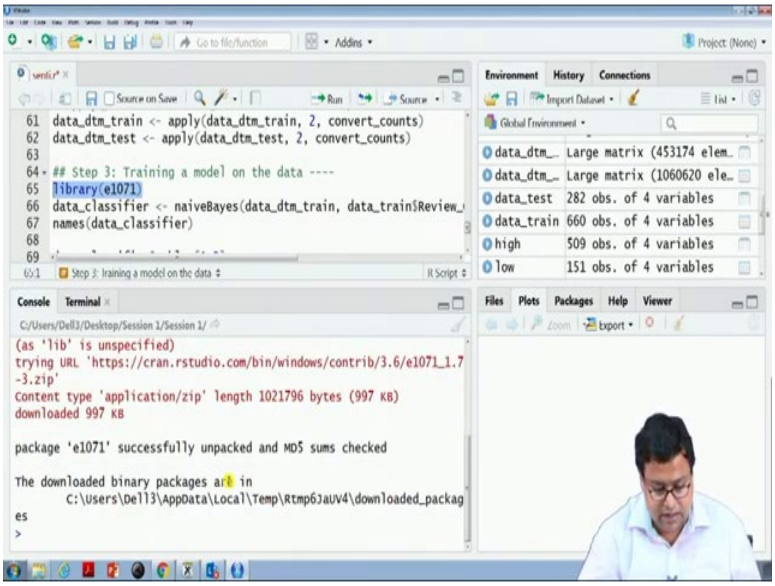
(Refer Slide Time: 26:08)



```
data_dtm_train <- apply(data_dtm_train, 2, convert_counts)
data_dtm_test <- apply(data_dtm_test, 2, convert_counts)

## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)
```

```
> library(e1071)
Error in library(e1071) : there is no package called 'e1071'
> install.packages('é1071')
Installing package into 'C:/Users/Dell3/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
Warning in install.packages :
  package 'é1071' is not available (for R version 3.6.1)
> install.packages('e1071')
Installing package into 'C:/Users/Dell3/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
```

data_dtm_...	Large matrix (453174 elem...
data_dtm_...	Large matrix (1060620 ele...
data_test	282 obs. of 4 variables
data_train	660 obs. of 4 variables
high	509 obs. of 4 variables
low	151 obs. of 4 variables

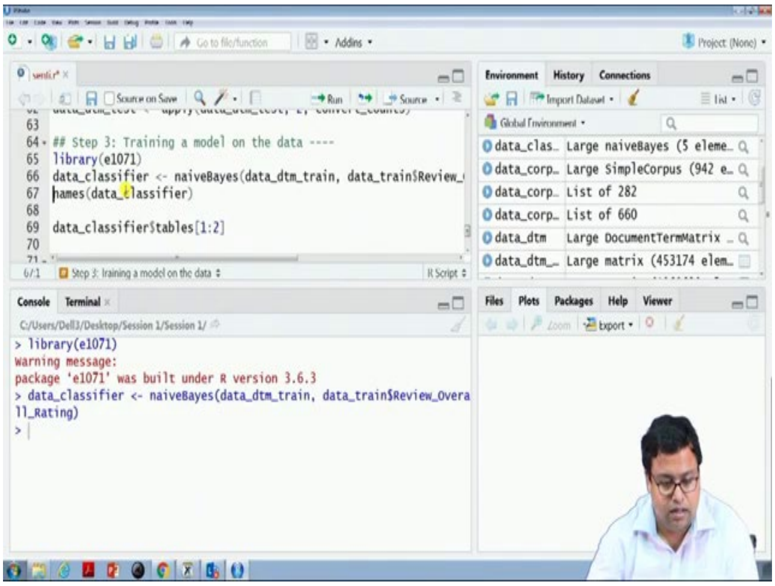


```
data_dtm_train <- apply(data_dtm_train, 2, convert_counts)
data_dtm_test <- apply(data_dtm_test, 2, convert_counts)

## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)
```

```
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/e1071_1.7
-3.zip'
Content type 'application/zip' length 1021796 bytes (997 KB)
downloaded 997 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Dell3\AppData\Local\Temp\Rtmp6JaUV4\downloaded_packag
es
>
```

So I will now use this document term matrix training to my model building. So the library that we will use for Naive Bayes algorithm is E1071 that is not there. So install packages E1071 oh sorry I should have written the e properly. Okay so it is installing E1071 right now it has installed it.

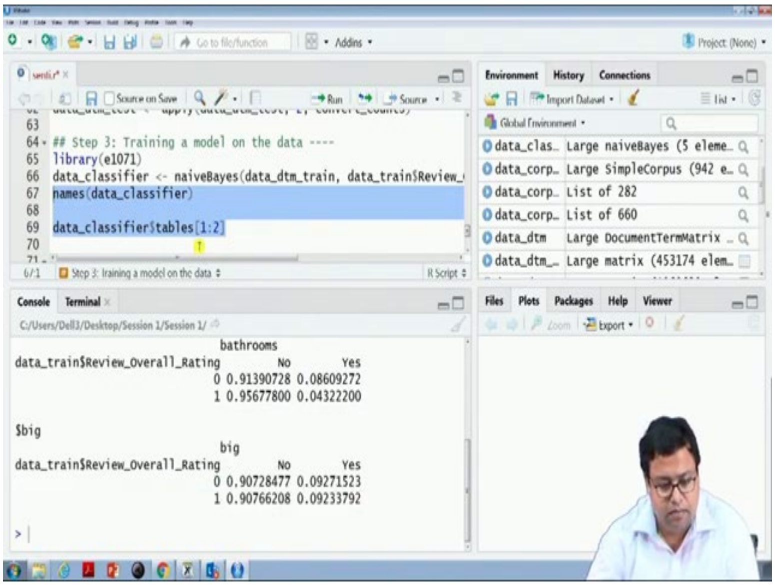
(Refer Slide Time: 26:48)



```
## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)

data_classifier$tables[1:2]
```

```
> library(e1071)
Warning message:
package 'e1071' was built under R version 3.6.3
> data_classifier <- naiveBayes(data_dtm_train, data_train$Review_Overa
ll_Rating)
> |
```

```
data_clas..  Large naiveBayes (5 eleme..
data_corp..  Large SimpleCorpus (942 e..
data_corp..  List of 282
data_corp..  List of 660
data_dtm     Large DocumentTermMatrix ..
data_dtm_..  Large matrix (453174 elem..
```



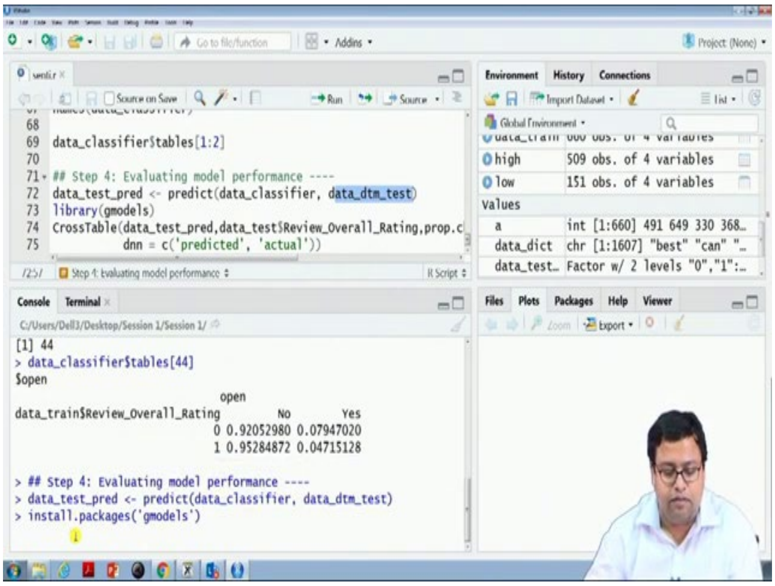
```
## Step 3: Training a model on the data ----
library(e1071)
data_classifier <- naiveBayes(data_dtm_train, data_train$Review_
names(data_classifier)

data_classifier$tables[1:2]
```

```
                              bathrooms
data_train$Review_Overall_Rating         No        Yes
                               0 0.91390728 0.08609272
                               1 0.95677800 0.04322200

$big
                              big
data_train$Review_Overall_Rating         No        Yes
                               0 0.90728477 0.09271523
                               1 0.90766208 0.09233792

> |
```

Now I will call this library it is okay warning message, but we do not care. Now _classifier the model name is Naive Bayes this is the algorithm name while this is my x axis which is the term matrix and the y variable is basically the overall rating that you got 1, 0 rating. So if I now run a simple this time I got the classifier.

And if I want to see the probabilities of the classifier I can see for at least first two you see bathrooms this is 0.91, 0.95 and this is 0.8 I do not think bathroom is something they are based on which things are changing a lot. Big is not at all changing, but why do not I find out group let us find out group what this group is.

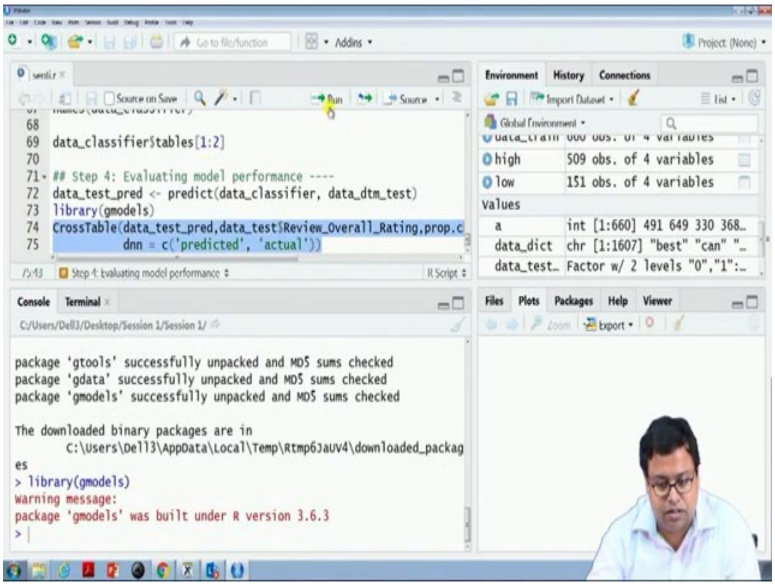
(Refer Slide Time: 27:38)



```
data_classifier$tables[1:2]

## Step 4: Evaluating model performance ----
data_test_pred <- predict(data_classifier, data_dtm_test)
library(gmodels)
CrossTable(data_test_pred,data_test$Review_Overall_Rating,prop.c
           dnn = c('predicted', 'actual'))
```

```
[1] 44
> data_classifier$tables[44]
$open
                              open
data_train$Review_Overall_Rating         No        Yes
                               0 0.92052980 0.07947020
                               1 0.95284872 0.04715128

> ## Step 4: Evaluating model performance ----
> data_test_pred <- predict(data_classifier, data_dtm_test)
> install.packages('gmodels')
```

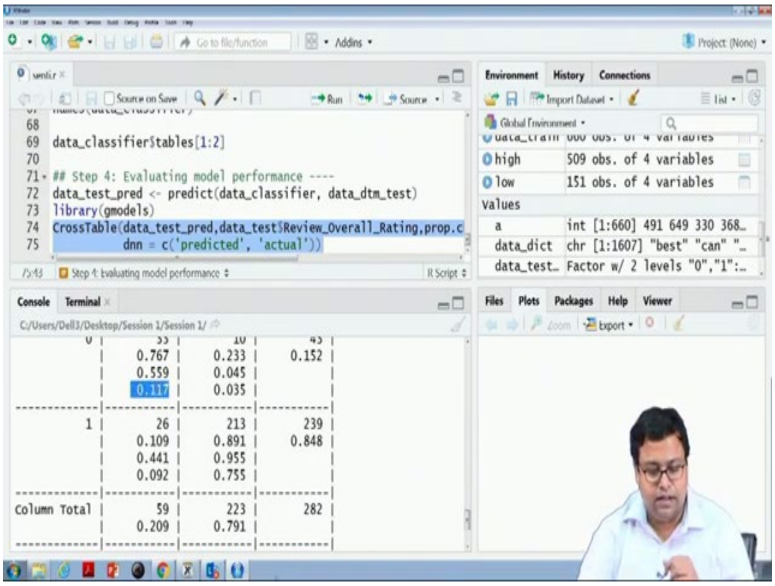
```
high    509 obs. of 4 variables
low     151 obs. of 4 variables
Values
a           int [1:660] 491 649 330 368…
data_dict   chr [1:1607] "best" "can" "…
data_test…  Factor w/ 2 levels "0","1":…
```



```
data_classifier$tables[1:2]

## Step 4: Evaluating model performance ----
data_test_pred <- predict(data_classifier, data_dtm_test)
library(gmodels)
CrossTable(data_test_pred,data_test$Review_Overall_Rating,prop.c
           dnn = c('predicted', 'actual'))
```

```

package ‘gtools’ successfully unpacked and MD5 sums checked
package ‘gdata’ successfully unpacked and MD5 sums checked
package ‘gmodels’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Dell3\AppData\Local\Temp\Rtmp6JaUV4\downloaded_packag
es
> library(gmodels)
Warning message:
package ‘gmodels’ was built under R version 3.6.3
> 
```

```
high    509 obs. of 4 variables
low     151 obs. of 4 variables
Values
a           int [1:660] 491 649 330 368…
data_dict   chr [1:1607] "best" "can" "…
data_test…  Factor w/ 2 levels "0","1":…
```



```
data_classifier$tables[1:2]

## Step 4: Evaluating model performance ----
data_test_pred <- predict(data_classifier, data_dtm_test)
library(gmodels)
CrossTable(data_test_pred,data_test$Review_Overall_Rating,prop.c
          dnn = c('predicted', 'actual'))
```

high	509 obs. of 4 variables	
low	151 obs. of 4 variables	
Values		
a	int [1:660] 491 649 330 368…	
data_dict	chr [1:1607] "best" "can" "…	
data_test…	Factor w/ 2 levels "0","1":…	

```
             |     0.767 |     0.233 |     0.152 |
             |     0.559 |     0.045 |           |
             |     0.117 |     0.035 |           |
-------------|-----------|-----------|-----------|
           1 |        26 |       213 |       239 |
             |     0.109 |     0.891 |     0.848 |
             |     0.441 |     0.955 |           |
             |     0.092 |     0.755 |           |
-------------|-----------|-----------|-----------|
Column Total |        59 |       223 |       282 |
             |     0.209 |     0.791 |           |
-------------|-----------|-----------|-----------|
```

So if I am not wrong I will find out group here good here and good is 44. So if I just see data .classifier 44 instead of 1 and 2. If I just try to see the 44th one that is the good things and you can see carefully that here when good is actually occurring yes then it is 8 % times this is a positive rating and 4 only 4.7 % case.

So there is a huge odd ratios difference 92 is to 8 divided 95 is to 4, so I can say that good is something which leads to classification. So we are doing it with the only the review content you can do the same stuff exactly same stuff with the review title.

So now I use this to predict my data here DTM dm testing data, testing document term matrix wee se for the prediction purpose and then I will call the library again the installed library installed packages. So see I am installing because today I am using a different system actually. If you have once installed in your system you do not have to install again. So library g-models and then I will do the cross tabulation.

So the cross tabulation is saying that I have my overall accuracy level is 213+33 which is around 246/282. So this many times I am overall correct 87 % of time based on this and if I try to focus on which is basically 75.5 + 117 this one. Now if I try to focus on that this is my actual this is my predicted. Now if I am trying to see that how out of actual ones how many times you have correct the predicted one which is basically 213/223 comes up to be 0.955.

So once positive rating I am properly identifying negative ratings if I just focus on then negative rating identification is not so good because there 59 observations which are negative basically, but I am currently predicting only 33 of them so which might not be good. So we have to try to improve this classification better using further models. So now this is an example how we can do text mining.

Now I can use this particular text mining technique which is this Naive Bayes to create another data term matrix and can predict that whether that is a positive rating or negative rating. So this is how basically we create a prediction easy prediction mechanism to predict the sentiment of a particular text. In the next session, we will discuss about various kinds of libraries that we can use for testing purpose. Thank you very much I will see you in the next video.