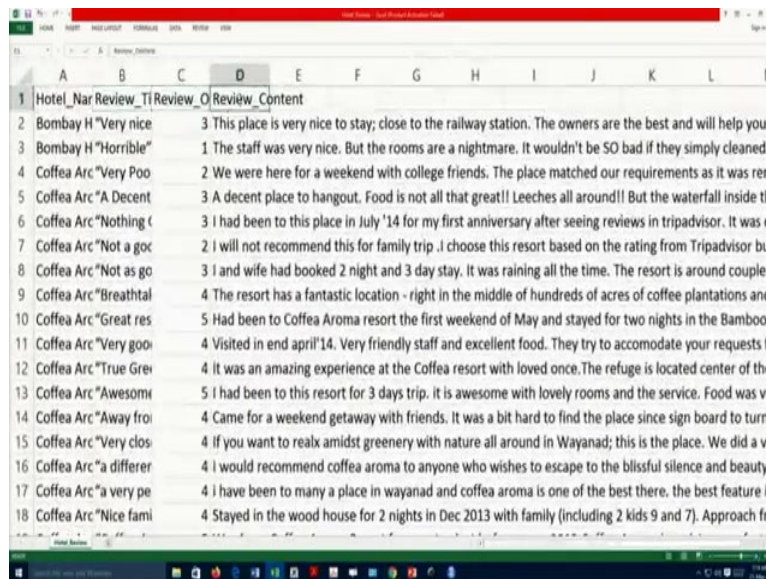


**Marketing Analytics**  
**Professor Swagato Chatterjee**  
**Vinod Gupta School of Management,**  
**Indian Institute of Technology, Kharagpur**  
**Lecture 58**  
**Text Mining and Sentiment Analytics (Contd.)**

Hello everybody. Welcome to Marketing Analytics course, this is Doctor Swagato Chatterjee from VGSOM, IIT Kharagpur who is taking this course for you. We are in currently in week 11 and this is session 3 and we are discussing about sentiment mining and Text Mining and Sentiment Analytics and we are trying to find out till the last class we have found out that we can use various kinds of lexicons which are libraries to get sentiment from a dataset.

So in this particular class we will actually do that in hand and then we will also try to find out certain other insights from this particular thing. So the dataset that I am going to use is the same old dataset which has been used in week 1. So let me just open the dataset.

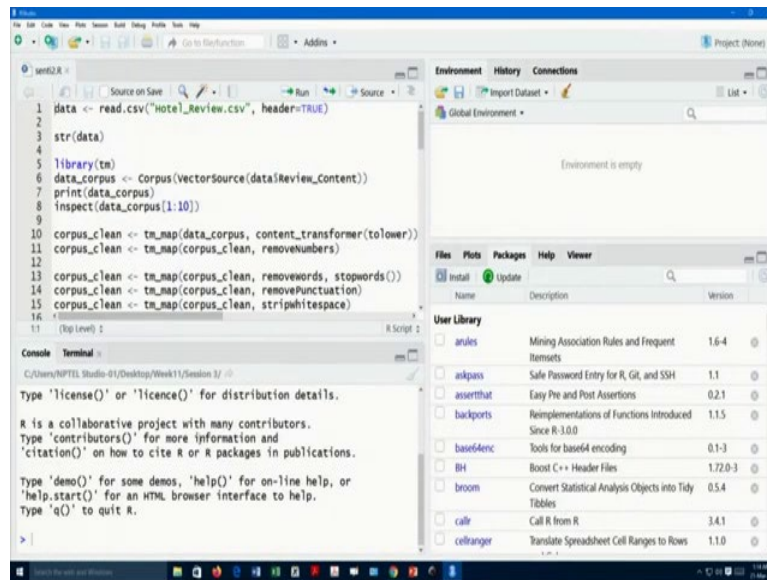
(Refer Slide Time: 01:03)



Hotel_Nar	Review_Ti	Review_Q	Review_Content
Bombay H	"Very nice"	3	This place is very nice to stay; close to the railway station. The owners are the best and will help you
Bombay H	"Horrible"	1	The staff was very nice. But the rooms are a nightmare. It wouldn't be SO bad if they simply cleaned.
Coffea Arc	"Very Poo"	2	We were here for a weekend with college friends. The place matched our requirements as it was ren
Coffea Arc	"A Decent"	3	A decent place to hangout. Food is not all that great!! Leeches all around!! But the waterfall inside th
Coffea Arc	"Nothing c"	3	I had been to this place in July '14 for my first anniversary after seeing reviews in tripadvisor. It was c
Coffea Arc	"Not a goc"	2	I will not recommend this for family trip .i choose this resort based on the rating from Tripadvisor bu
Coffea Arc	"Not as go"	3	I and wife had booked 2 night and 3 day stay. It was raining all the time. The resort is around couple
Coffea Arc	"Breathtal"	4	The resort has a fantastic location - right in the middle of hundreds of acres of coffee plantations and
Coffea Arc	"Great res"	5	Had been to Coffea Aroma resort the first weekend of May and stayed for two nights in the Bamboo
Coffea Arc	"Very goo"	4	Visited in end april'14. Very friendly staff and excellent food. They try to accomodate your requests f
Coffea Arc	"True Grei"	4	It was an amazing experience at the Coffea resort with loved once.The refuge is located center of the
Coffea Arc	"Awesomi"	5	I had been to this resort for 3 days trip. It is awesome with lovely rooms and the service. Food was ve
Coffea Arc	"Away froi"	4	Came for a weekend getaway with friends. It was a bit hard to find the place since sign board to turn
Coffea Arc	"Very clos"	4	if you want to relax amidst greenery with nature all around in Wayanad; this is the place. We did a vi
Coffea Arc	"a differer"	4	I would recommend coffea aroma to anyone who wishes to escape to the blissful silence and beauty
Coffea Arc	"a very pe"	4	I have been to many a place in wayanad and coffea aroma is one of the best there. the best feature i
Coffea Arc	"Nice fami"	4	Stayed in the wood house for 2 nights in Dec 2013 with family (including 2 kids 9 and 7). Approach fr

So this is the dataset that we have if you know the hotel review, the review title at the second column then the overall rating in the third column and the review content in the fourth column. So this is the same dataset which we will be using,

(Refer Slide Time: 01:19)



```
1 data <- read.csv("note_review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- Corpus(VectorSource(data$review_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17 >
```

Environment History Connections  
Global Environment  
Environment is empty

User Library

Name	Description	Version
arules	Mining Association Rules and Frequent Itemsets	1.6-4
askpass	Safe Password Entry for R, Git, and SSH	1.1
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.5
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.72.0-3
broom	Convert Statistical Analysis Objects into Tidy Tibbles	0.5.4
callr	Call R from R	3.4.1
cellranger	Translate Spreadsheet Cell Ranges to Rows	1.1.0

And I have senti ,, senti2.R which is another R-file with which we will be doing our work in this particular session. The first job in any kind of this thing the first job is to read the dataset. So we first will read the dataset to read the dataset the first thing that I have to do is to set my working directory to source file location.

(Refer Slide Time: 01:46)

```
1 data <- read.csv("Hotel_Review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- Corpus(VectorSource(data$Review_Content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17 (Dip Level) 1
```

Environment History Connections  
Global Environment  
Data  
data 942 obs. of 4 variables

Console Terminal  
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >  
> setwd("c:/users/NPTEL Studio-01/Desktop/week11/session 3")  
> data <- read.csv("Hotel\_Review.csv", header=TRUE)  
>

```
1 data <- read.csv("Hotel_Review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- Corpus(VectorSource(data$Review_Content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17 (Dip Level) 1
```

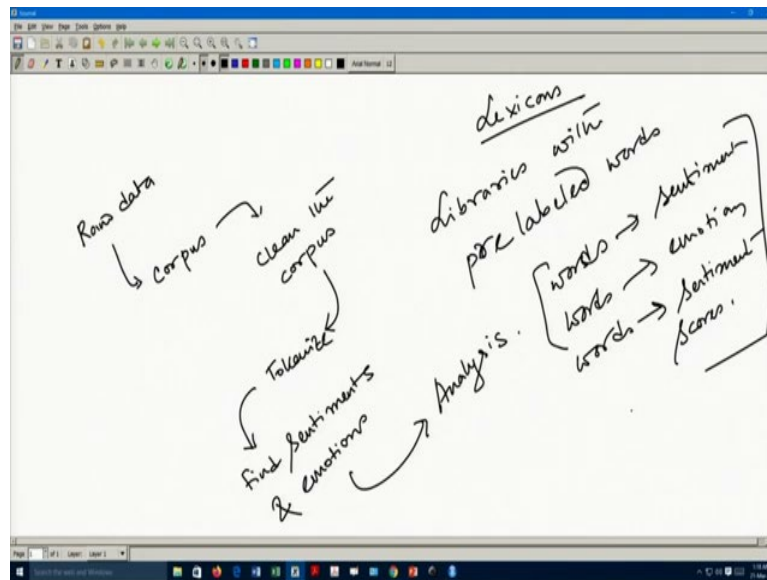
Environment History Connections  
Global Environment  
Data  
data 942 obs. of 4 variables

Console Terminal  
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >  
> 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 ...  
 \$ Review\_Title : Factor w/ 923 levels ""\ COMFORTABLE & GOOD TO STAY IN RANTH  
 AMBORE 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 "I- Location:- superb location; just can  
 e from Hotel). Nice c\liff view;- service;- PATHETIC; they wont respect yo"| \_\_\_\_truncated\_  
 ...: 689 626 911 9 211 317 185 613 136 747 ...  
>

So this is something that I am doing. And then the second step is to read the dataset, so data is = read.csv, read.csv is a code that reads the csv file and the hotel dataset is hotel review.csv file which is the same csv file which has kept in session 3 and which we have also used in session 1 also of this week and then header = true. So I run it once I have run it we have to check the structure of the data.

So the structure of the data says that you see there is a review title and there is a review hotel name city and there is a review title. Hotel name city is about the name of the hotel and review title is about the title of this particular review and then review is overall rating and then review content so these kind of things are there some basic things are there. So then what we will do?

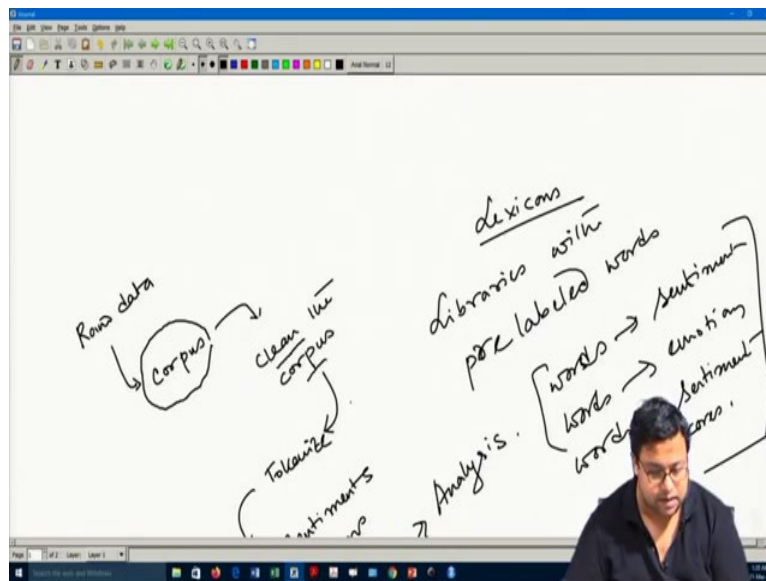
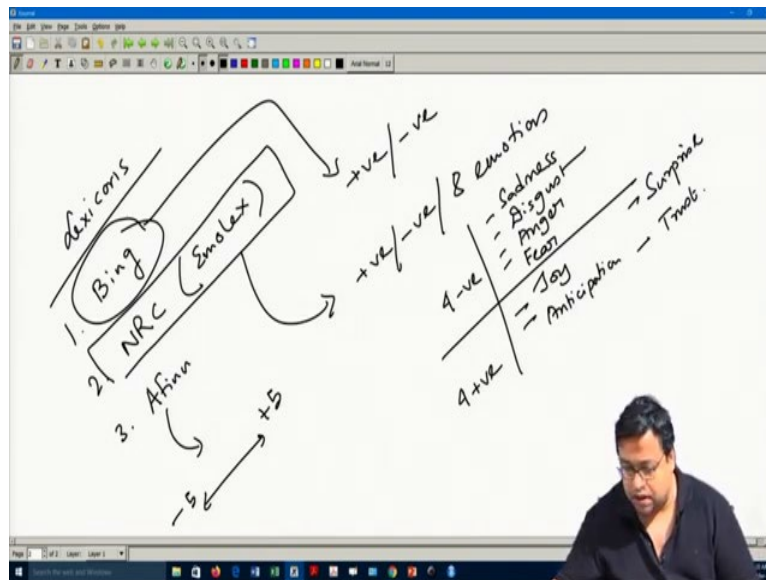
(Refer Slide Time: 03:01)



The first job is in any data analysis, the first job is that you got the raw data from there you have to convert it to a corpus from there you have to clean the corpus. So after cleaning the corpus you have to let us say do data mining in this case tokenize and then for each token that means each word is a token find sentiments and emotions and with this we will do some analysis. So this is our steps that are involved.

Now if I want to know that what I am going to do here is that we remember we will be using lexicons. Lexicons are basically libraries with pre-labeled words. So pre-labeled words, words are either has been labeled as sentiment or has been labeled as emotions or has been scored as sentiments, so sentiment scores. These are the 3 choices that we have and for that we have 3 lexicons basically.

(Refer Slide Time: 04:45)

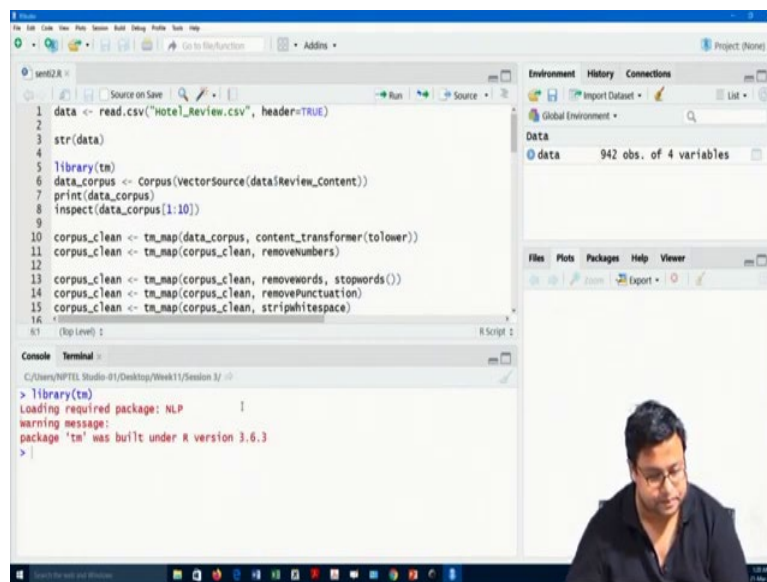


If you remember in the last class we discussed there are lexicons so 3 lexicons. Lexicon number 1 is Bing, lexicon number 2 is NRC also known as Emolex and lexicon number 3 is Afinn. So Bing only says whether it is positive or negative that is all. This guy says positive or negative and then 8 emotions and this guy says - 5 to + 5 in this continuum whatever score do you have.

And then this 8 emotions are there are 4 negative emotions and 4 positive emotions the negative emotions are basically sadness and then disgust and then anger and fear these are the and in this case there are 8 classic emotion the basic emotions are like joy and then probably anticipation and then here it is surprise and the last one joy, anticipation, surprise and the fourth one will be let us say delight oh no trust.

The fourth one if I am not wrong fourth one is trust. So these are the 8 basic emotions based on which the scores are given. If my focus is only to find out positive sentiment or negative sentiment I will use Bing. If my focus is to use the emotion scores also we will use this NRC library or Emolex library and that is what we are going to do now. So I have read the data so as I told before after reading the data I have to change it to corpus. And I have to clean the corpus so this is something that I am going to do now.

(Refer Slide Time: 07:11)



```
1 data <- read.csv("note_Review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- Corpus(VectorSource(data$Review_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removenumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removewords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removepunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripwhitespace)
16
17 # (tip level) :
```

Console Terminal

```
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 11 />
> library(tm)
Loading required package: NLP
warning message:
package 'tm' was built under R version 3.6.3
>
```

So the library that will be required is tm, so I call this tm library.

(Refer Slide Time: 07:16)

```
1 data <- read.csv("note_Review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- Corpus(VectorSource(data$review_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17
18
19
20
21
22
23
```

Environment History Connections  
Global Environment  
Data  
data 942 obs. of 4 variables  
data\_corpusLarge SimpleCorpus (942 e. Q

Files Plots Packages Help Viewer  
Zoom Export

```
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >
> data_corpus <- Corpus(VectorSource(data$review_content))
> print(data_corpus)
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 942
>
```

```
1 data <- read.csv("note_Review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- Corpus(VectorSource(data$review_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17
18
19
20
21
22
23
```

Environment History Connections  
Global Environment  
Data  
data 942 obs. of 4 variables  
data\_corpusLarge SimpleCorpus (942 e. Q

Files Plots Packages Help Viewer  
Zoom Export

```
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >
> data_corpus <- Corpus(VectorSource(data$review_content))
> print(data_corpus)
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 942
>
[7] I and wife had booked 2 night and 3 day stay. It was raining all the time. The res
ort is around couple of kilometers to the top from main road. They will take you to the
top. Food was very basic. Also nice dal, chappathi and sabzi which looked li
ke dal. since it was raining lot of leeches around the resort. Had to be careful all t
he while. People were friendly and very nice. But seriously nothing great to write about
it. Place looked mostly empty. Advice: pay bit more and go for vythri village; vythri
resort or any other established ones..
```

And here I am changing it to my corpus and data dollar review content which is the text of the review is my dataset with which I will play. So data\_corpus = corpus and then vectors of data dollar review content and I read the, and this corpus of 942 elements has been created.

So if I want to print the corpus it will just say that I have 942 documents right now and if I want to inspect the documents that is the first 10 is the same old score and these are my first 10 documents. So I and my wife had booked 2 night which is the 7 document and 3 days stay it was raining all time. The resort is around couple of kilometers to the top of the main road blah, blah, blah. So this kind of thing is there, fair enough.

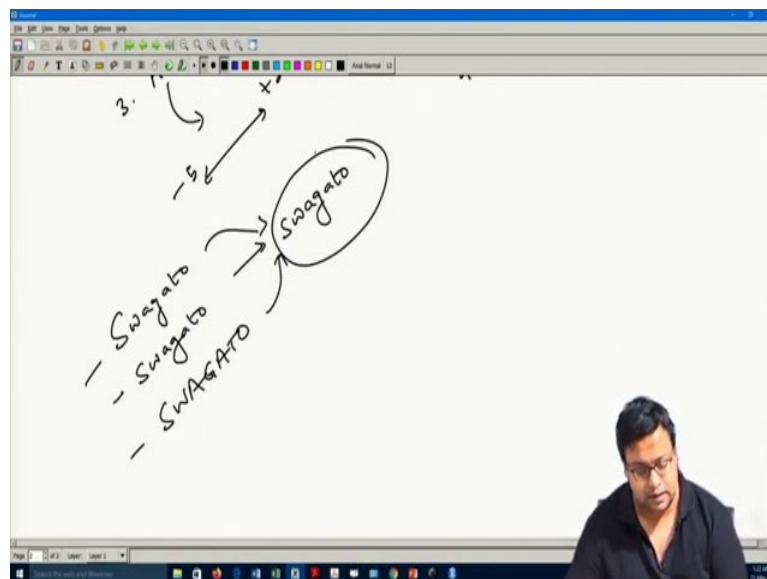
(Refer Slide Time: 08:17)

```
1 data <- read.csv("note_Review.csv", header=TRUE)
2
3 str(data)
4
5 library(tm)
6 data_corpus <- corpus(VectorSource(data$Review_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripwhitespace)
16
101 (Tip Level) 1
```

Environment History Connections  
Global Environment  
Data  
data 942 obs. of 4 variables  
data\_corpusLarge SimpleCorpus (942 e...

Files Plots Packages Help Viewer  
Zoom Export

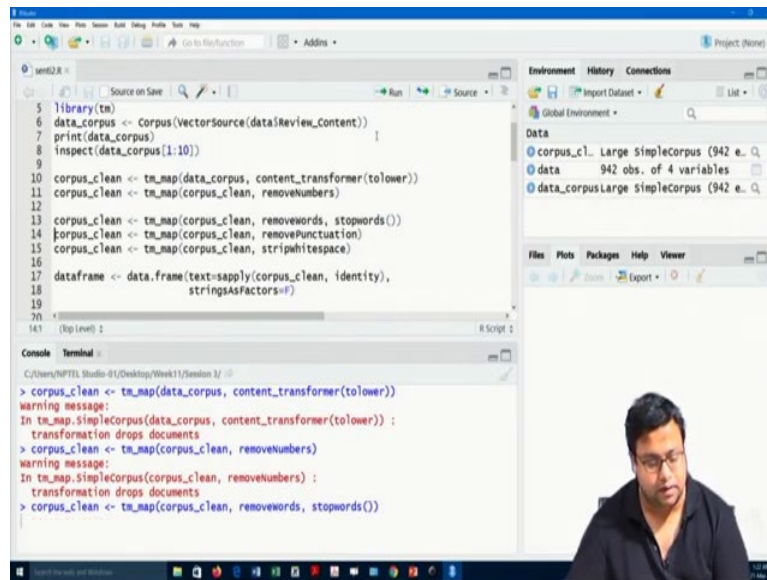
Console Terminal  
C:/Users/NPTEL Studio/Desktop/Week11/Session 3/ ->



So now what the next job is I have to change it to the lower means all the text I will change it to their lower case. Why will I change it to the lower case? Yes, so that this 2 things are understood equally. So let us say somebody says and somebody says these 3 things should be same all will be converted to this will be the case. So I have to make sure that these 3 guys are considered as a same word so that is why I will change it to lower.



(Refer Slide Time: 09:06)



The screenshot shows the RStudio interface with a script editor on the left and a console on the bottom. The script editor contains the following R code:

```
5 library(tm)
6 data_corpus <- Corpus(VectorSource(dataReview_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18                          stringsAsFactors=F)
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

The console shows the following output:

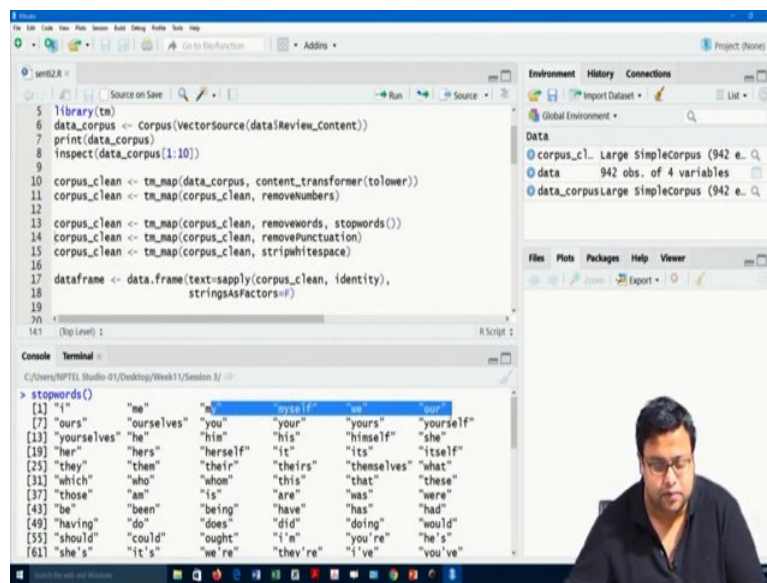
```
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ > corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
warning message:
In tm_map.SimpleCorpus(data_corpus, content_transformer(tolower)) :
transformation drops documents
> corpus_clean <- tm_map(corpus_clean, removeNumbers)
warning message:
In tm_map.SimpleCorpus(corpus_clean, removeNumbers) :
transformation drops documents
> corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
```

The Environment pane on the right shows the following objects:

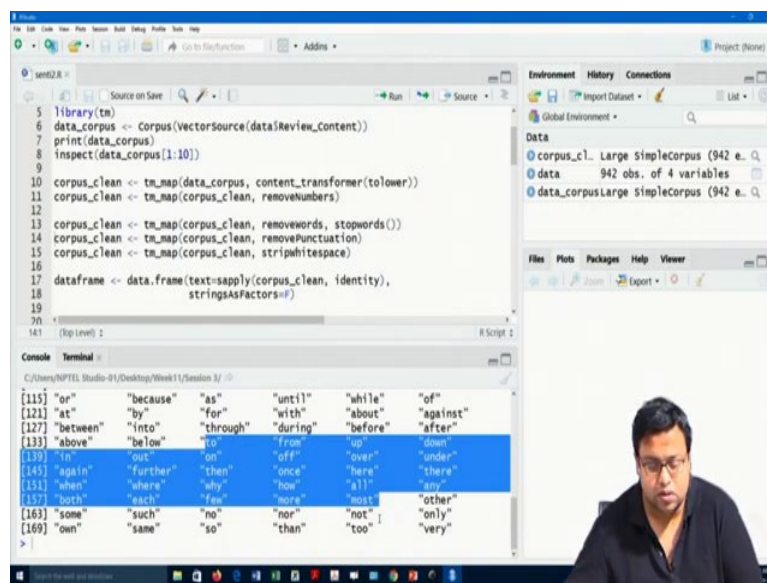
- corpus\_c1: Large SimpleCorpus (942 e...)
- data: 942 obs. of 4 variables
- data\_corpus: Large SimpleCorpus (942 e...)

Then I remove the numbers from the dataset the whole the corpus, from whole corpus I am removing the numbers then I am removing the stop words also fair enough. What are the stop words that are currently there?

(Refer Slide Time: 09:23)



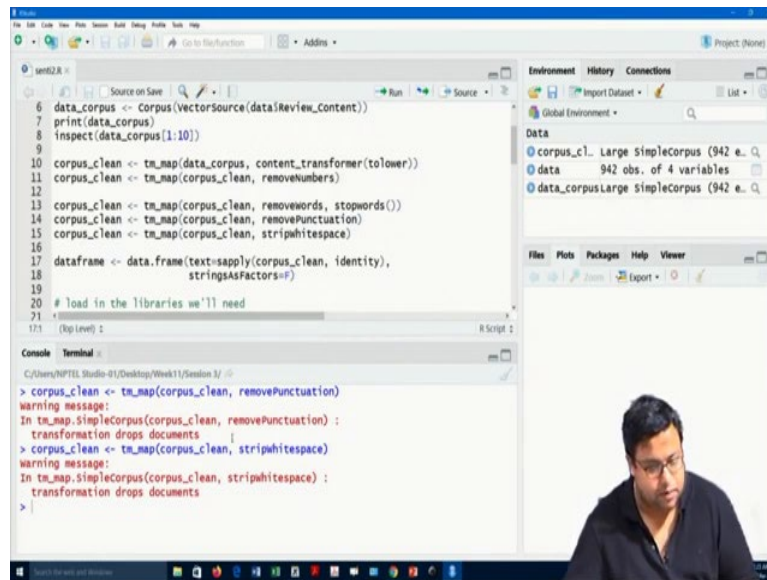
```
5 library(tm)
6 data_corpus <- Corpus(VectorSource(dataReview_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(to_lower))
11 corpus_clean <- tm_map(corpus_clean, remove_numbers)
12
13 corpus_clean <- tm_map(corpus_clean, remove_words, stopwords())
14 corpus_clean <- tm_map(corpus_clean, remove_punctuation)
15 corpus_clean <- tm_map(corpus_clean, strip_whitespace)
16
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18 stringsAsFactors=F)
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



```
115 "or" "because" "as" "until" "while" "of"
121 "at" "by" "for" "with" "about" "against"
127 "between" "into" "through" "during" "before" "after"
133 "above" "below" "to" "from" "up" "down"
139 "in" "on" "off" "over" "under"
145 "again" "further" "then" "once" "here" "there"
151 "when" "where" "why" "how" "all" "any"
157 "both" "each" "few" "more" "most" "other"
163 "some" "such" "no" "nor" "not" "only"
169 "own" "same" "so" "than" "too" "very"
```

If I just write stop words and like this, so these are the stop words like I, me, mine, myself, we, our, yourself, these, there some of the other things like while, through, of, how. So there can be some pronoun, some conjunctions, some WH words all of these things are your stop words which has been removed from the dataset.

(Refer Slide Time: 09:54)



```
6 data_corpus <- Corpus(VectorSource(dataReview_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(tolower))
11 corpus_clean <- tm_map(corpus_clean, removeNumbers)
12
13 corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripWhitespace)
16
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18                          stringsAsFactors=F)
19
20 # load in the libraries we'll need
21
22 (top level) :
```

Environment History Connections  
Global Environment  
Data  
corpus\_c1. Large SimpleCorpus (942 e...  
data 942 obs. of 4 variables  
data\_corpusLarge SimpleCorpus (942 e...  
Files Plots Packages Help Viewer  
Export

```
> 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
>
```

I removed the punctuation also and I remove the white space also. So once I remove all of these things this dataset has become clean data. Now I have to convert this clean data through understanding this dataset is ultimately is a corpus which is a text data. And there are some tm library can work with this kind of a corpus not other libraries can do that. So some other libraries require some other kind of dataset requirement and a very basic kind of dataset requirement in text data is the character form.

So you have change it to basic raw character form so I have to do that again. I have to change it to basic raw character form.

(Refer Slide Time: 10:35)

```
6 data_corpus <- Corpus(VectorSource(dataReview_content))
7 print(data_corpus)
8 inspect(data_corpus[1:10])
9
10 corpus_clean <- tm_map(data_corpus, content_transformer(to_lower))
11 corpus_clean <- tm_map(corpus_clean, remove_numbers)
12
13 corpus_clean <- tm_map(corpus_clean, remove_words, stopwords())
14 corpus_clean <- tm_map(corpus_clean, remove_punctuation)
15 corpus_clean <- tm_map(corpus_clean, strip_whitespace)
16
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18                         stringsAsFactors=F)
19
20 # Load in the libraries we'll need
21
22 [1] (skip level)
23
24 Error: unexpected numeric constant in "corpus_clean$1"
25 > inspect(corpus_clean[1])
26 <<SimpleCorpus>>
27 Metadata: corpus specific: 1, document level (indexed): 0
28 Content: documents: 1
29
30 [1] place nice stay close railway station owners best will help questions definitely r
31 ecommend place teej festival place neighbourhood can celebrate
32 > dataframe <- data.frame(text=sapply(corpus_clean, identity),
33                           stringsAsFactors=F)
34 >
```

```
> inspect(corpus_clean[1])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1
[1] place nice stay close railway station owners best will help questions definitely r
2 ecommend place teej festival place neighbourhood can celebrate
3 will recommend family trip choose resort based rating tripod...
4 decent place hangout food great leeches around waterfall in...
5 place july first anniversary seeing reviews tripod/r stars li...
6 will recommend family trip choose resort based rating tripod...
7 wife booked night day stay raining time resort around coupl...
8 resort fantastic location right middle hundreds acres coffee ...
9 coffee aroma resort first weekend may stayed two nights ba...
Showing 1 to 9 of 942 entries
```

If I see corpus clean dollar and then let us say 1. If I just so wait a minute if I just try to see the how the corpus clean looks like. It is a large simple corpus and the content is the first content is this. The second content is this where the content is there and correspondingly the other details are also there. So if I want to inspect the corpus clean first element it will look like this kind of a text. So what we are trying to do right now is we will be applying a identity on the corpus clean.

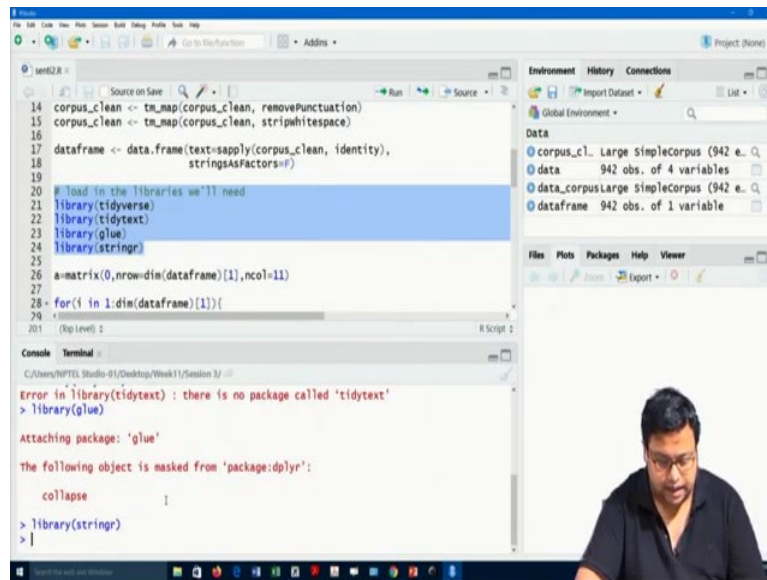
So what is the identity function? Identity function will only take the text of it. Identity function will take the text of it and put in this text column and then strings as factors so change the strings to the factor this is false. So do not when you convert to a data frame do

not consider the strings to be factors keep it strings that is keep it characters that is how I am changing the dataset.

So now that I have changed it I got this thing. So here I got a data frame of 942 observations where each observation each cell is a clean data file which is a clean text file and there is no punctuation there, no capital numbers are there and anything is there. So the data frame is data frame and if I just say that okay first row this is looks like this, this is the first one. Second row will look like this.

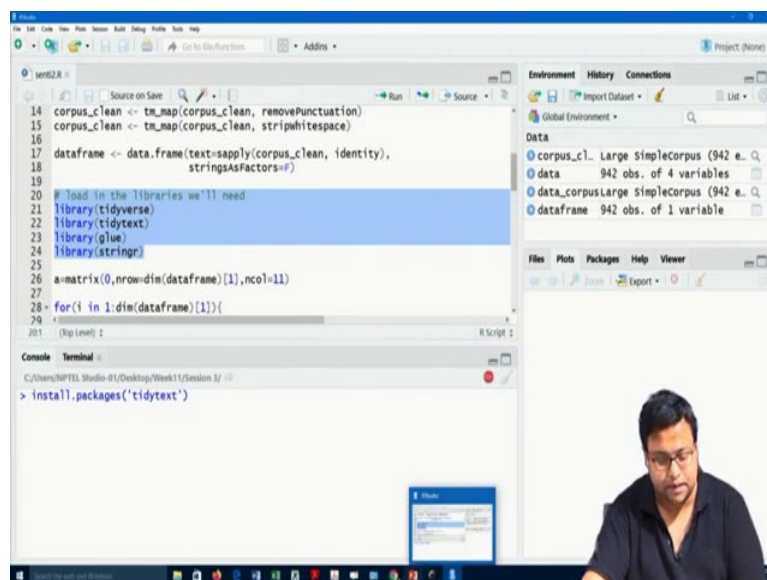
So each a text which no space, no blank space more than one space bars are not there, no numbers are there, no punctuation is there, no stop words are there, clean corpus value is what I am getting. Now what does this NRC and etcetera we will be doing?

(Refer Slide Time: 13:14)



```
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripwhitespace)
16
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18 stringsAsFactors=F)
19
20 # load in the libraries we'll need
21 library(tidyverse)
22 library(tidytext)
23 library(glue)
24 library(stringr)
25
26 a=matrix(0,nrow=dim(dataframe)[1],ncol=11)
27
28 for(i in 1:d1m(dataframe)[1]){
29
30 }
31 }
32 }
```

```
Console Terminal
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >
Error in library(tidytext) : there is no package called 'tidytext'
> library(glue)
Attaching package: 'glue'
The following object is masked from 'package:dplyr':
collapse
> library(stringr)
> |
```



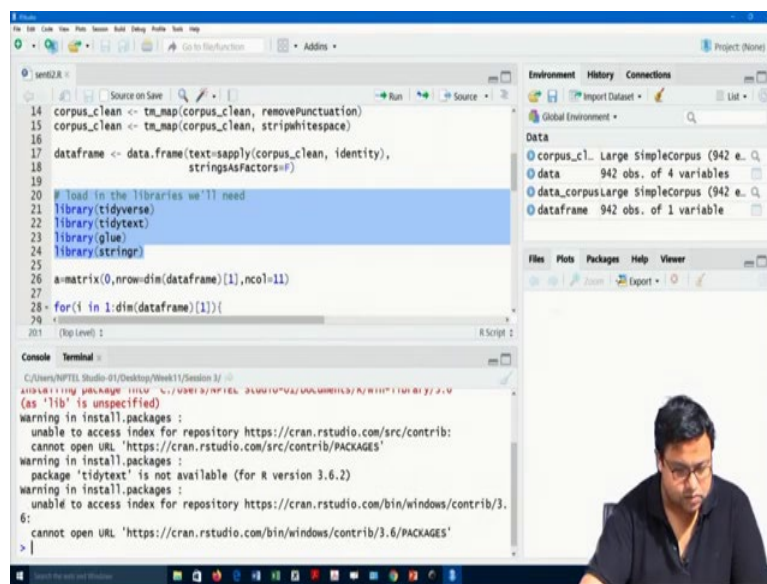
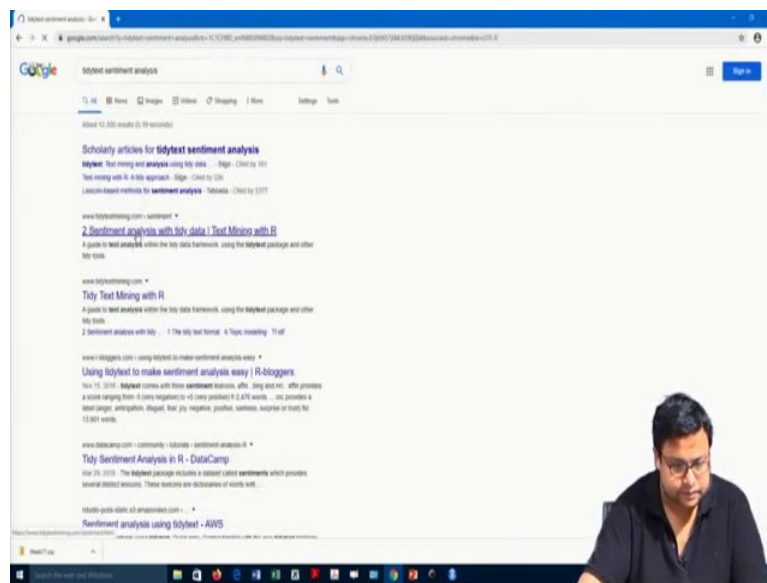
```
14 corpus_clean <- tm_map(corpus_clean, removePunctuation)
15 corpus_clean <- tm_map(corpus_clean, stripwhitespace)
16
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18 stringsAsFactors=F)
19
20 # load in the libraries we'll need
21 library(tidyverse)
22 library(tidytext)
23 library(glue)
24 library(stringr)
25
26 a=matrix(0,nrow=dim(dataframe)[1],ncol=11)
27
28 for(i in 1:d1m(dataframe)[1]){
29
30 }
31 }
32 }
```

```
Console Terminal
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >
> install.packages('tidytext')
```

NRC to for example let us say Bing. Bing as I told Bing is a library which talks about positive and negative. So if I just call the libraries first, so let us these are the libraries that we will need so I will call the library one by one.

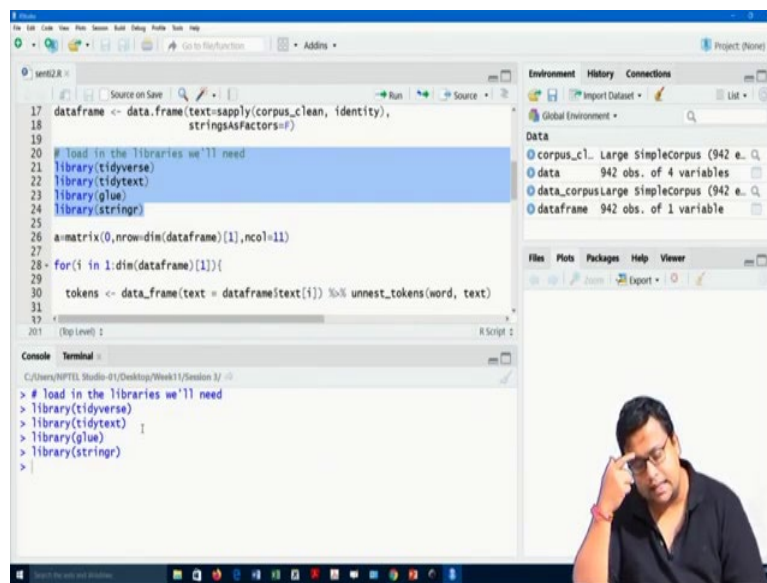
Okay so there are some library which are not there like tidy text is not there and tidy text is only not there. So I have to install tidy text so install packages and then tidy text. So let it get installed. So tidy text is the in the last class we are discussing in terms of this particular link we were discussing that. In that link we found the tidy text is the library from where this NRC and etcetera things are kept.

(Refer Slide Time: 14:16)



For example if I just open the link tidy text sentiment analysis okay so let me install once this. So till now we have just converted the corpus file to the character file and it is a clean version of the character as I just have shown you. So right now what we will do is we will try to find out what are the sentiments in each of these things.

(Refer Slide Time: 15:07)

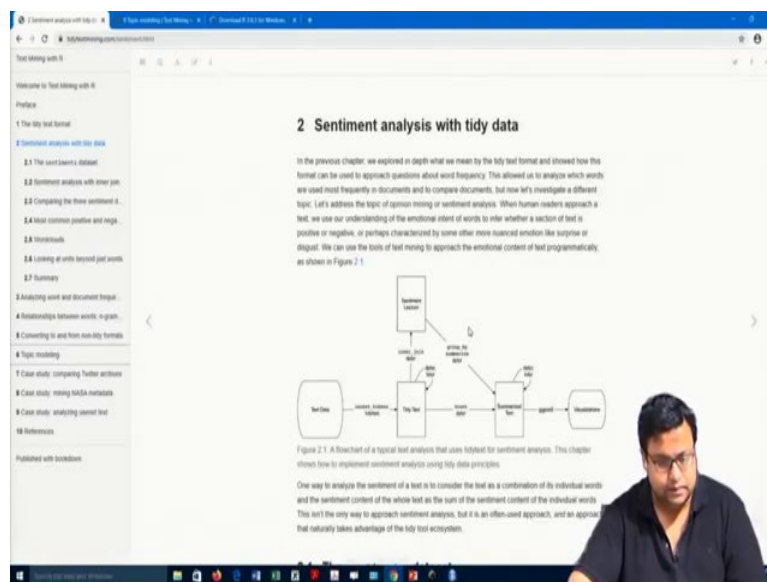


```
17 dataframe <- data.frame(text=sapply(corpus_clean, identity),
18 stringsAsFactors=F)
19
20 # Load in the libraries we'll need
21 library(tidyverse)
22 library(tidytext)
23 library(glue)
24 library(stringr)
25
26 a=matrix(0,nrow=dim(dataframe)[1],ncol=11)
27
28 for(i in 1:dim(dataframe)[1]){
29
30 tokens <- data_frame(text = dataframe$text[i]) %>% unnest_tokens(word, text)
31
32 }
33
34 }
```

The screenshot shows the RStudio interface. The script editor on the left contains the code above. The console on the bottom left shows the execution of the library loading commands. The environment pane on the right shows the loaded objects: corpus\_c1, data, data\_corpusLarge, and dataframe.

So the library that we will be required is these 4 libraries. So I will call them so tidy text is the same library the thing that we are discussing in the last class about the library that has been created and we were discussing in this website how tidy text can help.

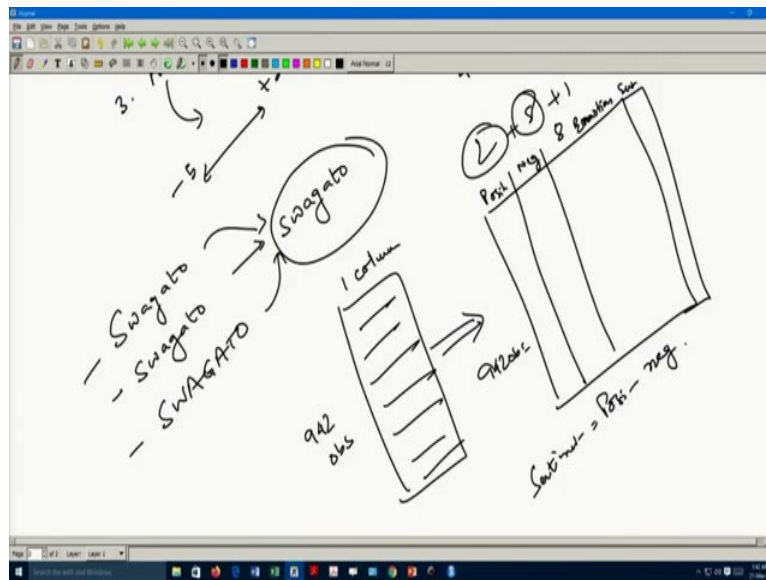
(Refer Slide Time: 15:27)



So we have discussed all of this thing in the last class. So this is something that we are using. Now in that tidy text will work for our version 3.6.3. So if you have a earlier version please download and install the latest version of our and then you call this library. So next what we will do is see what is our purpose here, what is the thing that I am going to do here?



(Refer Slide Time: 15:56)



```
25 a=matrix(0,nrow=dim(dataframe)[1],ncol=11)
26
27
28 for(i in 1:dim(dataframe)[1]){
29
30 tokens <- data_frame(text = dataframe$text[i]) %>% unnest_tokens(word, text)
31
32 # get the sentiment from the first text:
33 b=as.numeric(tokens %>%
34 inner_join(get_sentiments("nrc")) %>% # pull out only sentiment words
35 count(sentiment) %>% # count the # of positive & negative words
36 spread(sentiment, n, fill = 0) %>%
37 mutate(pos=ifelse(exists("positive"),positive,0),
38 neg=ifelse(exists("negative"),negative,0),
39 trust=ifelse(exists("trust"),trust,0),
40
282 (Dip Level)
R Script
Console Terminal
C:/Users/NPTEL/Studio-01/Desktop/Week11/Session 3/ > dim(dataframe)
[1] 942 1
> dim(dataframe)[1]
[1] 942
>
```

So I have a data frame with 942 observations and one column each column is one text. From here I want this particular columns 942 observation again I want the positive sentiment how many words are there in positive sentiment.

How many words are there in negative sentiment and if I am using it for Emolex then all the 8 emotions each of this 8 emotions how many words are there this is what I want and then rather I want the n total sentiments. Total sentiment I can write it as sentiment = probably positive - negative. So something like that is what I am trying, so here how many columns are there? 2 columns here, 8 columns here and then last one sentiment column, so total 11 columns are there.

So that is why you are creating right now a blank matrix  $A = \text{matrix}(0, n, \text{row}, n)$ ,  $n$  row means number of rows that is  $\text{dim}(\text{data frame } 1)$ . What is  $\text{dim}(\text{data frame } 1)$ ? So let us just write down what is  $\text{dim}$  of data frame,  $\text{dim}$  of data frame means the dimension of data frame which is giving me 942, 1 that means this data frame has 942 rows and 1 column.

$\text{Dim}(\text{data frame } 1)$  is only the first entry of this thing which is 942. So basically I am writing that I am getting a matrix which is 942 rows and 11 columns, 11 columns why because one is positive, one is negative then 8 emotions and then the total overall sentiments all of these things are there. So the values will be the count of words that means in document 1 how many words are there in positive sentiment that will be in the positive one.

How many words are there which is associated with negative sentiment that will go to the negative one and then how many words are associated with anger, fear blah, blah, blah all of these things will be populated. So I am creating a matrix like that then what I am doing is I am running a for loop, so this is a for loop which starts from here and ends here. So let us just assume to understand what this thing is.

(Refer Slide Time: 18:20)

```
27  
28 for(i in 1:dim(dataframe)[1]){  
29  
30 tokens <- data_frame(text = dataframe$text[i]) %>% unnest_tokens(word, text)  
31  
32 # get the sentiment from the first text:  
33 bmas.numeric(tokens %>%  
34 inner_join(get_sentiments("nrc")) %>% # pull out only sentiment words  
35 count(sentiment) %>% # count the # of positive & negative words  
36 spread(sentiment, n, fill = 0) %>%  
37 mutate(pos=ifelse(exists("positive"),positive,0),  
38 neg=ifelse(exists("negative"),negative,0),  
39 trust=ifelse(exists("trust"),trust,0),  
40 joy=ifelse(exists("joy"),joy,0),  
41 antic=ifelse(exists("anticipation"),anticipation,0),  
42  
43 }  
44  
45 }  
46  
47  
48 }  
49  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }
```

Environment History Connections  
Global Environment  
Data  
corpus\_c1. Large SimpleCorpus (942 e...  
data 942 obs. of 4 variables  
data\_corpusLarge SimpleCorpus (942 e...  
dataframe 942 obs. of 1 variable  
tokens 20 obs. of 1 variable

Console Terminal  
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >  
> i=1  
> dataframe\$text[i]  
[1] " place nice stay close railway station owners best will help questions definitely r  
ecommend place teej festival place neighbourhood can celebrate"  
> tokens <- data\_frame(text = dataframe\$text[i]) %>% unnest\_tokens(word, text)  
> tokens

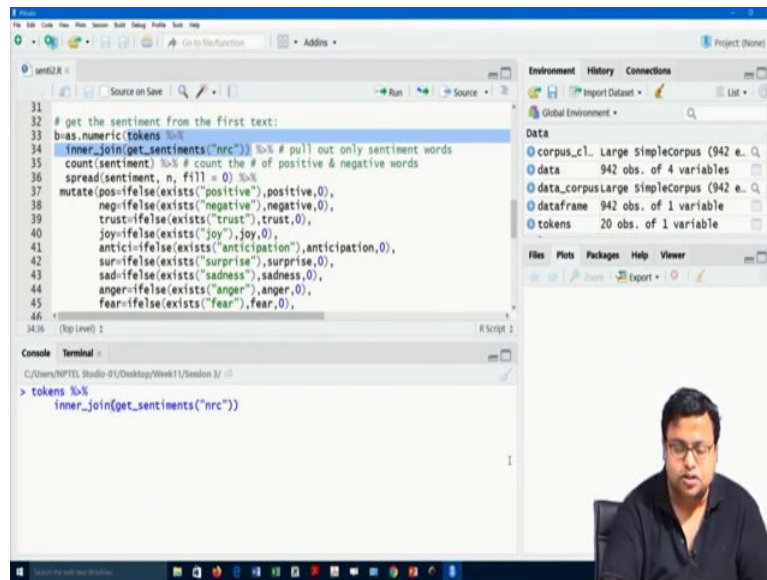
```
11 questions  
12 definitely  
13 recommend  
14 place  
15 teej  
16 festival  
17 place  
18 neighbourhood  
19 can  
20 celebrate  
>
```

Environment History Connections  
Global Environment  
Data  
corpus\_c1. Large SimpleCorpus (942 e...  
data 942 obs. of 4 variables  
data\_corpusLarge SimpleCorpus (942 e...  
dataframe 942 obs. of 1 variable  
tokens 20 obs. of 1 variable

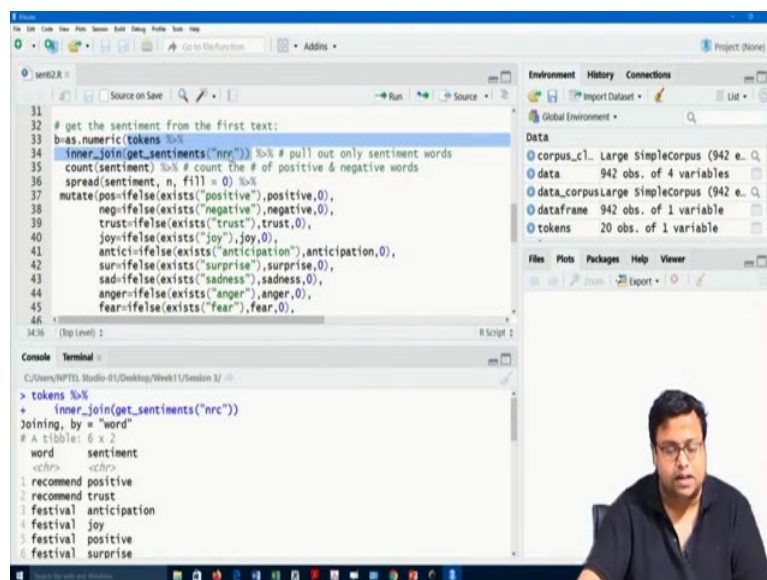
Let us just assume that  $i = 1$ , so  $i =$  varies from 1 to 942 dim data frame 1 means 942 so  $i$  is varying from 1 to 942, but let us assume  $i = 1$  what then is happening. When  $i = 1$  then this is basically the first text, the text of the first entry. So what I am doing is I am creating tokens for those text. So word wise tokens so I am just running this.

So if I am running this what are the tokens, the tokens are basically they are breaking this thing into words. So place, nice, stay, close, railway station, owners so I can create tokens as even sentences also or something like that also. So here we are creating word wise token each word is one token. So I have 20 into 1 that means 20 unique words are there in this particular text.

(Refer Slide Time: 19:23)



```
31 # get the sentiment from the first text:
32
33 # get the sentiment from the first text:
34 b=as.numeric(tokens %>%
35   inner_join(get_sentiments("nrc")) %>% # pull out only sentiment words
36   count(sentiment) %>% # count the # of positive & negative words
37   spread(sentiment, n, fill = 0) %>%
38   mutate(pos=ifelse(exists("positive"),positive,0),
39           neg=ifelse(exists("negative"),negative,0),
40           trust=ifelse(exists("trust"),trust,0),
41           joy=ifelse(exists("joy"),joy,0),
42           antici=ifelse(exists("anticipation"),anticipation,0),
43           sur=ifelse(exists("surprise"),surprise,0),
44           sad=ifelse(exists("sadness"),sadness,0),
45           anger=ifelse(exists("anger"),anger,0),
46           fear=ifelse(exists("fear"),fear,0),
47           )
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



```
> tokens %>%
+   inner_join(get_sentiments("nrc"))
Joining, by = "word"
# A tibble: 6 x 2
  word sentiment
<-chr> <-chr>
1 recommend positive
2 recommend trust
3 festival anticipation
4 festival joy
5 festival positive
6 festival surprise
```

Then what I am doing, just check this part that I am using one by one. So I will just run this much okay copy this much and pasted it here. So I am picking up tokens and then inner joins sentiments from NRC. So if I run this it will just say that out of all this words that I have in my tokens in my tokens which words are having these are the words out of this 20 words which words are associated with certain sentiment or emotions from this NRC library.

Now first time you run this, first time you run this NRC it will ask that why do not you install a library called text data. So you have to again install a package called text data. Then again you run it, it will say that okay there is a researcher called Saif Mohammed who has created this library so he will say that okay before you can use this library why do not you give me the credit for this particular thing.

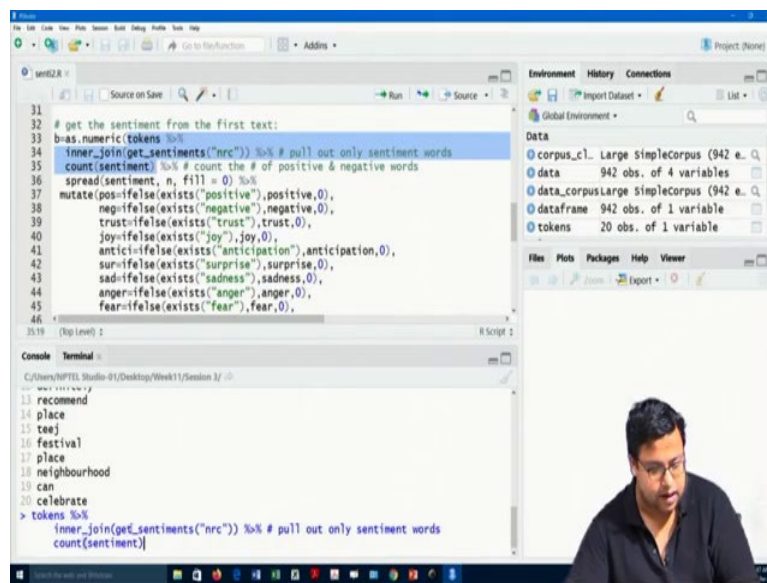
So you have to write 1 and then press enter so you will face this thing I am telling you. So once you press 1 and enter then 127 MB file will get downloaded which is actually the library. So after that gets downloaded only you can run this. So I am not doing that in this particular video because that will take time. In my case it is already downloaded so now I am using this NRC library.

So NRC library has 8 emotions and 2 sentiment positive and negative. So after that what I am doing is I am joining it and in this join what I am getting is for every word corresponding sentiment and emotion is getting. So for example here there is a word called recommend which is basically a positive sentiment and that emotion associated with this word is trust. Similarly, there is a festival which is a positive sentiment and the emotions that are associated with this word is anticipation.

So we anticipate what will happen in a festival we wait for Diwali or Durga Puja then the joy and then a surprise. So these are the certain I would say emotions and sentiments that are associated with certain words. So for first entry only I found basically 2 words, 2 positive words and 3 sentiments or 4 sentiments probably each has a count 1. So positive has a count 2 this and this.

Trust has a count 1, anticipation has a count 1, joy has a count 1 and surprise has a count 1. So other guys are not counted then what I am doing. I will be running one step ahead.

(Refer Slide Time: 22:18)

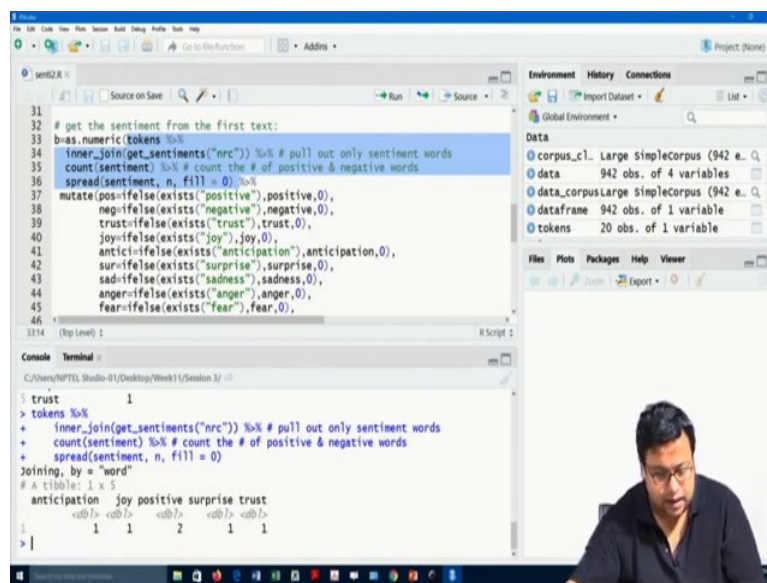


```
31 # get the sentiment from the first text:
32
33 b=as.numeric(tokens %>%
34   inner_join(get_sentiments("nrc")) %>% # pull out only sentiment words
35   count(sentiment) %>% # count the # of positive & negative words
36   spread(sentiment, n, fill = 0) %>%
37   mutate(pos=ifelse(exists("positive"),positive,0),
38          neg=ifelse(exists("negative"),negative,0),
39          trust=ifelse(exists("trust"),trust,0),
40          joy=ifelse(exists("joy"),joy,0),
41          antic=ifelse(exists("anticipation"),anticipation,0),
42          sur=ifelse(exists("surprise"),surprise,0),
43          sad=ifelse(exists("sadness"),sadness,0),
44          anger=ifelse(exists("anger"),anger,0),
45          fear=ifelse(exists("fear"),fear,0),
46
```

Environment History Connections  
Global Environment  
Data  
corpus\_cl. Large simplecorpus (942 e...  
data 942 obs. of 4 variables  
data\_corpusLarge Simplecorpus (942 e...  
dataframe 942 obs. of 1 variable  
tokens 20 obs. of 1 variable

Files Plots Packages Help Viewer

Console Terminal  
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >  
13 recommend  
14 place  
15 teej  
16 festival  
17 place  
18 neighbourhood  
19 can  
20 celebrate  
> tokens %>%  
inner\_join(get\_sentiments("nrc")) %>% # pull out only sentiment words  
count(sentiment)



```
5 trust 1
> tokens %>%
+ inner_join(get_sentiments("nrc")) %>% # pull out only sentiment words
+ count(sentiment) %>% # count the # of positive & negative words
+ spread(sentiment, n, fill = 0)
Joining, by = "word"
# A tibble: 1 x 5
  anticipation joy positive surprise trust
  <dbl> <dbl> <dbl> <dbl> <dbl>
1 1 1 2 1 1
>
```

Environment History Connections  
Global Environment  
Data  
corpus\_cl. Large simplecorpus (942 e...  
data 942 obs. of 4 variables  
data\_corpusLarge Simplecorpus (942 e...  
dataframe 942 obs. of 1 variable  
tokens 20 obs. of 1 variable

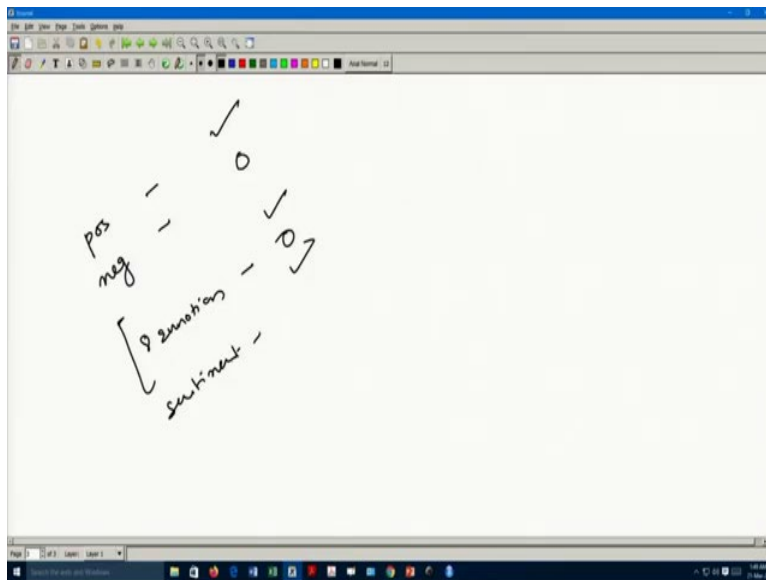
Files Plots Packages Help Viewer

Console Terminal  
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >  
5 trust 1  
> tokens %>%  
+ inner\_join(get\_sentiments("nrc")) %>% # pull out only sentiment words  
+ count(sentiment) %>% # count the # of positive & negative words  
+ spread(sentiment, n, fill = 0)  
Joining, by = "word"  
# A tibble: 1 x 5  
anticipation joy positive surprise trust  
<dbl> <dbl> <dbl> <dbl> <dbl>  
1 1 1 2 1 1  
>

Now this guy will give count up to this point if I just copy and paste and run this is giving me the count. They are saying that anticipating there is one word, joy there is one word, positive there is 2 words as I told you.

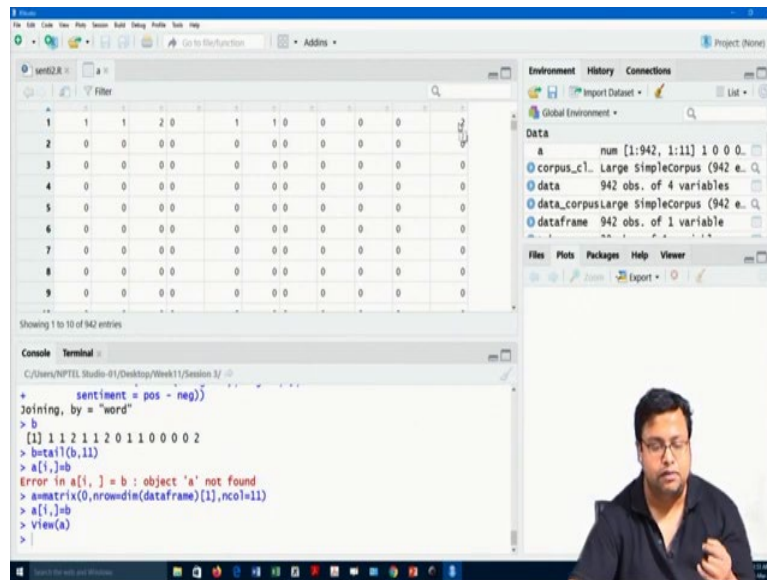
Then what I am doing then I am spreading it up, spreading it up means I am making it looking like this anticipation, joy, positive, surprise, trust and corresponding count. So whatever was the view previously I am just converting the view in a row wise. Now comes the next step now remember we with this thing what we are trying to do we will be mutating means changing this particular thing. Such that I have how many positive I need all the 11 counts I need the positive count.

(Refer Slide Time: 23:10)



```
26 a=matrix(0,nrow=dim(dataframe)[1],ncol=11)
27
28 for(i in 1:dim(dataframe)[1]){
29   tokens <- data_frame(text = dataframe$text[i]) %>% unnest_tokens(word, text)
30
31   # get the sentiment from the first text:
32   has.numeric(tokens %>%
33     inner_join(get_sentiments("nrc")) %>% # pull out only sentiment words
34     count(sentiment) %>% # count the # of positive & negative words
35     spread(sentiment, n, fill = 0) %>%
36     mutate(pos=ifelse.exists("positive"),positive,0),
37            neg=ifelse.exists("negative"),negative,0),
38            trust=ifelse.exists("trust"),trust,0),
39            joy=ifelse.exists("joy"),joy,0),
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
Console Terminal
C:/Users/NPTEL Studio-01/Desktop/Week11/Session 3/ >
+   anger=ifelse.exists("anger"),anger,0),
+   fear=ifelse.exists("fear"),fear,0),
+   dis=ifelse.exists("disgust"),disgust,0),
+   sentiment = pos - neg)
Joining, by = "word"
# A tibble: 1 x 14
  anticipation joy positive surprise trust pos neg antici sur sad anger
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1     1     1     2     1     1     2     0     1     1     0     0
# ... with 3 more variables: fear <dbl>, dis <dbl>, sentiment <dbl>
>
```



I need the positive count, the negative count, the 8 emotions basically and sentiment all of this things I need. So by chance if some sentiments are missing they should come as 0 by chance some sentiments are there they should come as corresponding counts. So this count should be there this count should be 0 in this particular case.

So how I can ensure that, that this count will be 0. To ensure that I am writing this particular function you see I am writing pos = if else function this is a if else function which says by chance positive exists. What does this positive exists mean? For example let us say in this particular thing do you think anything called let us say B or A, A or B is not here right now or let us say my name Swagato.

There is nothing in my name Swagato so if I just say exists Swagato it will say false there is nothing called Swagato in my global environment, but if I say exists and then write data if I write say 2 or if I write exist data frame it will say true because there is a data here there is a data frame here. Similarly if I say exist here positive. Positive is there which is a value of 2 so then I am saying that if positive exists then put positive whatever corresponding value otherwise 0.

For example negative does not exist so the neg this value should be 0. Trust exist so trust should be whatever the value of trust has come up which is 1, but joy does not exist oh joy also exist, but let us say sadness does not exist so corresponding value should be 0. So I am manually populating all the values based on the check that whether there particular sentiment is existing in this particular document or not.



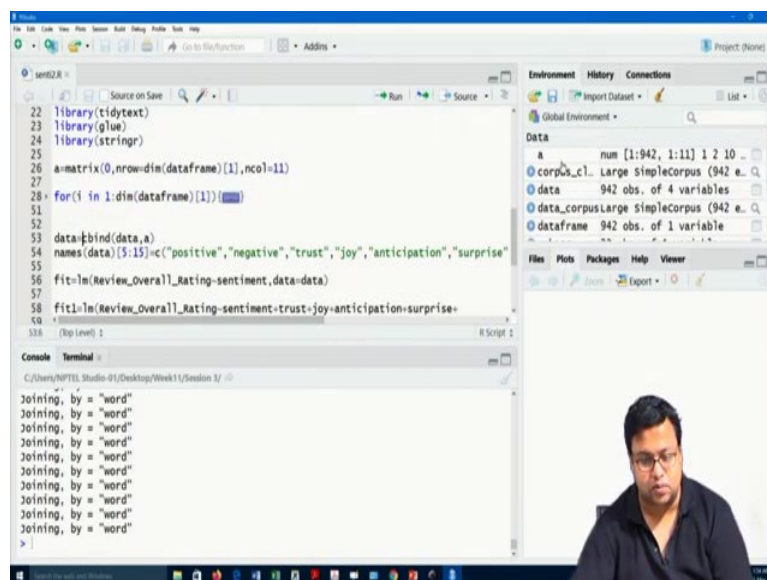
And last sentiment = positive - negative. So everything I am doing so now if I run this much if I run this much and populate it I will say that so this was a initial and this was joy, surprise up to this point you have got from the previous calculation then the rest like positive, negative, anticipation, surprise, sadness, anger and there were 3 more variables which has not been printed here these are the calculated part.

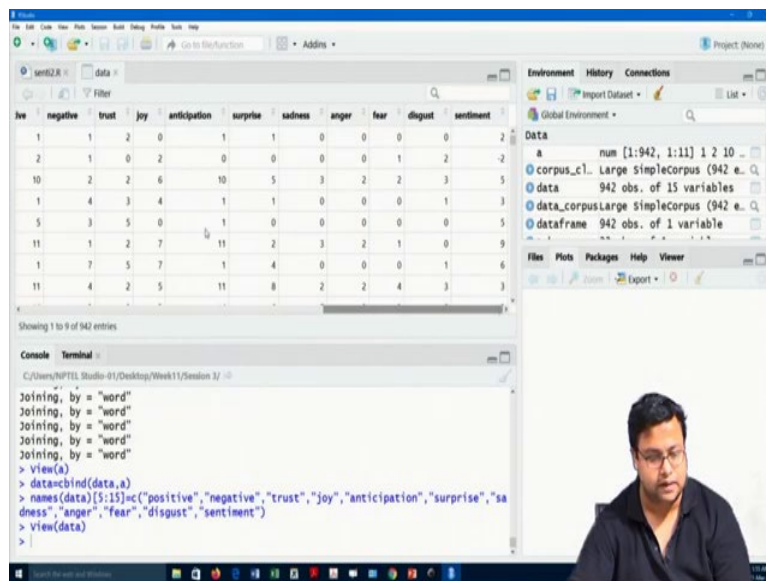
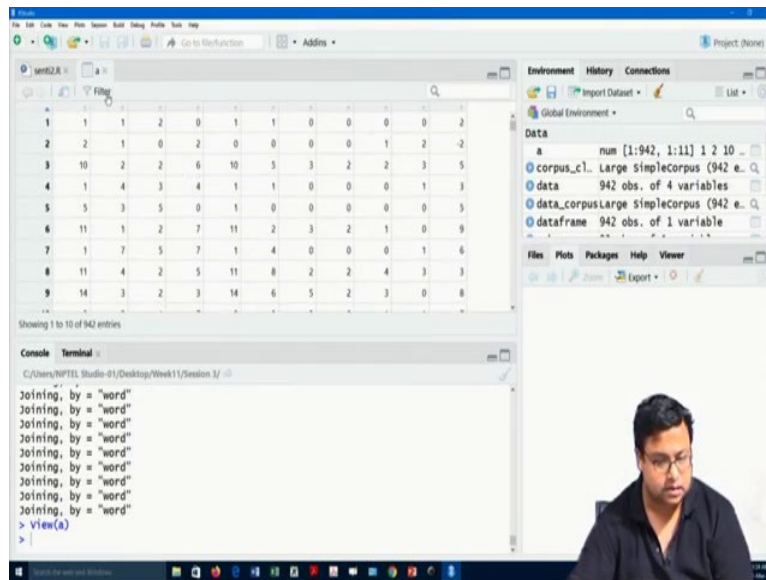
So because there is nothing negative n e g has come 0 because there is nothing sadness initially s a d has come 0 and so on. So I change this thing to a numeric variable and save it at b. So what is b then b looks like this 1, 1, 2, 1, 1. Now remember the last 11 values which is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 up to this, this values are something that is meaningful to me. Last 11 values are meaningful to me the rest of the values are not so meaningful to me.

So this is something that I will be picking up the last 11 values. Why 11? Because there are positive, negative then trust, joy, anticipation, surprise, sad, anger, fear and let us say disgust there are 8 emotions and then 1 sentiment. So 5, another 4 9 and 2 11 so these are my basic values based on which I will be trying to calculate everything. So that is what I am taking up till 11, b = b tail means take 11 values and populate it in ai.

That means in the first row of a okay so a I have not created. So in the first row of a because i = 1 populated that means right now the a has become like this everything is 0 the firstly it got populated the values are coming up. Now I will run this for the whole loop.

(Refer Slide Time: 27:44)





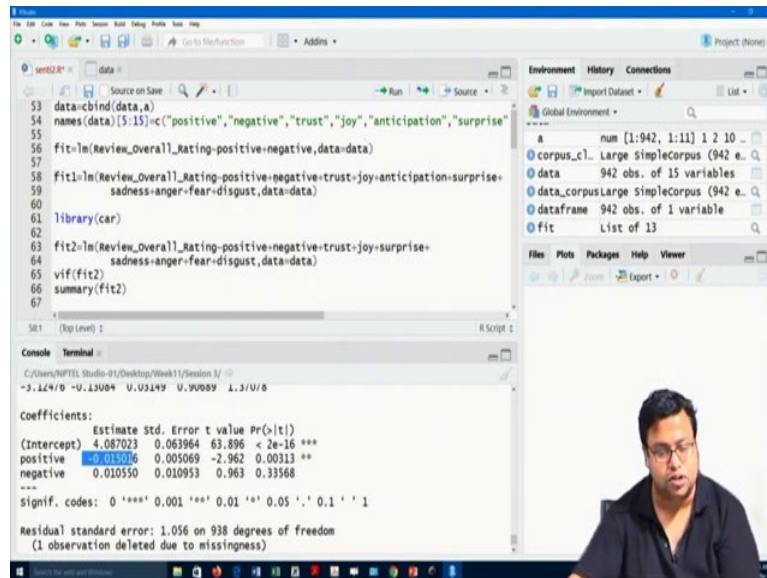
So if I run this for the whole loop, I will select from here I will select the whole loop and run it. So you can check where is the count here. Right now if I just refresh  $i = 82$  i will run from 1 to 942 so just keep on checking this number. So it will run from 1 to 942 each one of them it will pick up calculate based on NRC library what is coming up as your various kinds of emotions and sentiments and then it will populate it in A. Now the next job is to join this A with the original dataset to see that how emotions and sentiments is driving your rating behavior.

So how that emotions and sentiments expressed in the text will impact your overall rating is something that we are trying to do. So let us see now I will bind this A which is this values where there are sentiments and emotions all are populated here. I will bind this one just 1

minute yeah I will bind this one with the dataset. So I have bounded it fair enough and then the dataset has become like this.

The review title, overall rating, content and then positive, negative etcetera and sentiments those values are here so those values are given. So next what will I do if I got this values next what will I do?

(Refer Slide Time: 29:43)



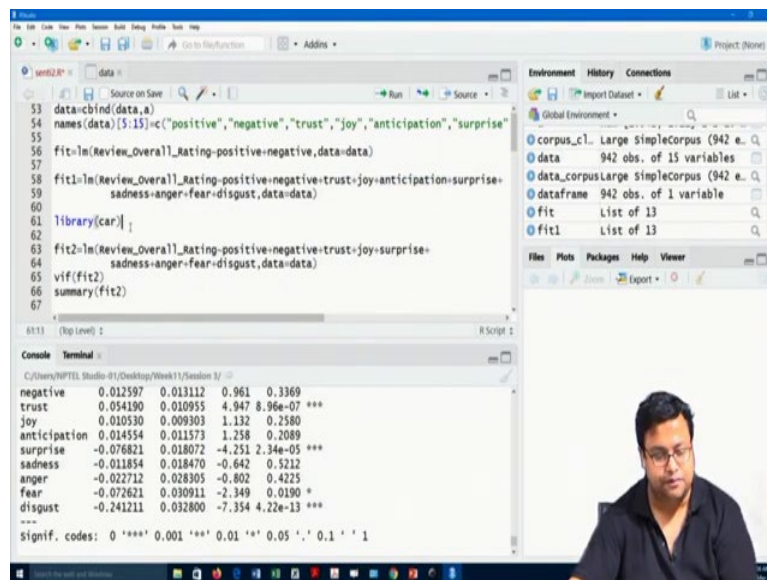
```
data <- data.frame(
  review_title = "A",
  overall_rating = 1,
  content = "This is a great product, I love it.",
  positive = "positive",
  negative = "negative",
  trust = "trust",
  joy = "joy",
  anticipation = "anticipation",
  surprise = "surprise",
  sadness = "sadness",
  anger = "anger",
  fear = "fear",
  disgust = "disgust"
)

fit1 <- lm(review_overall_rating ~ positive + negative, data = data)
summary(fit1)

library(car)
fit2 <- lm(review_overall_rating ~ positive + negative + trust + joy + surprise +
  sadness + anger + fear + disgust, data = data)
summary(fit2)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.087023	0.063964	63.896	< 2e-16 ***
positive	0.015006	0.005069	-2.962	0.00313 **
negative	0.010550	0.010953	0.963	0.33568

Residual standard error: 1.056 on 938 degrees of freedom  
(1 observation deleted due to missingness)



```
library(car)
fit2 <- lm(review_overall_rating ~ positive + negative + trust + joy + surprise +
  sadness + anger + fear + disgust, data = data)
summary(fit2)
```

	Estimate	Std. Error	t value	Pr(> t )
negative	0.012597	0.013112	0.961	0.3369
trust	0.054190	0.010955	4.947	8.96e-07 ***
joy	0.010530	0.009303	1.132	0.2580
anticipation	0.014554	0.011573	1.258	0.2089
surprise	-0.076821	0.018072	-4.251	2.34e-05 ***
sadness	-0.011854	0.018470	-0.642	0.5212
anger	-0.022712	0.028305	-0.802	0.4225
fear	-0.072621	0.030911	-2.349	0.0190 *
disgust	-0.241211	0.032800	-7.354	4.22e-13 ***

I will just go and find out how this thing is impacted by let us say positive + negative. How negative and positive sentiment is impacting me and how the various emotions are impacting my overall rating.

So first thing is positive and negative and then the emotions. So let us see first positive and negative. So if I run this and then if I try to see summary of it, it is giving me that positive ratings are highly associated with my codes, but negative are not so much. So my overall rating is highly I can say being explained by my positive ratings, but this is coming negative which is something we have to check first.

This should not come negative we have to check this code once again. Now next thing is I am trying to check the emotions let us say along with that. So I have come up with emotions and summary of it. If I see then certain emotions summary of fit 1 sorry summary of fit 1 then certain emotions are also associated with it. For example disgust and fear as you feel disgust or fear your rating goes down. Even sometimes as you feel trust your rating goes up something like that.

(Refer Slide Time: 31:10)

```
53 data=cbind(data,a)
54 names(data)[5:15]=c("positive","negative","trust","joy","anticipation","surprise")
55
56 fit=lm(Review_Overall_Rating~positive~negative,data=data)
57
58 fit1=lm(Review_Overall_Rating~positive~negative~trust~joy~anticipation~surprise~
59 sadness~anger~fear~disgust,data=data)
60
61 library(car)
62 vif(fit1)
63
64 fit2=lm(Review_Overall_Rating~positive~negative~trust~joy~surprise~
65 sadness~anger~fear~disgust,data=data)
66 vif(fit2)
67 summary(fit2)
68
```

Console Terminal

```
C:/Users/NPTEL/Studio-01/Desktop/Week11/Session 3/ >
Residual standard error: 0.9361 on 930 degrees of freedom
(1 observation deleted due to missingness)
Multiple R-squared: 0.2286, Adjusted R-squared: 0.2203
F-statistic: 27.55 on 10 and 930 DF, p-value: < 2.2e-16

> library(car)
Error in library(car) : there is no package called 'car'
> vif(fit1)
Error in vif(fit1) : could not find function "vif"
> install.packages("car")
```

```

53 data=cbind(data,a)
54 names(data)[5:15]=c("positive","negative","trust","joy","anticipation","surprise")
55
56 fit=lm(Review_Overall_Rating~positive+negative,data=data)
57
58 fit2=lm(Review_Overall_Rating~positive+negative+trust+joy+anticipation+surprise+
59 sadness+anger+fear+disgust,data=data)
60
61 library(car)
62 vif(fit1)
63
64 fit2=lm(Review_Overall_Rating~positive+negative+trust+joy+surprise+
65 sadness+anger+fear+disgust,data=data)
66 vif(fit2)
67 summary(fit2)
68
69 # (Tip Level) 1

```

```

53 data=cbind(data,a)
54 names(data)[5:15]=c("positive","negative","trust","joy","anticipation","surprise")
55
56 fit=lm(Review_Overall_Rating~positive+negative,data=data)
57
58 fit2=lm(Review_Overall_Rating~positive+negative+trust+joy+anticipation+surprise+
59 sadness+anger+fear+disgust,data=data)
60
61 library(car)
62 vif(fit1)
63
64 fit2=lm(Review_Overall_Rating~positive+negative+trust+joy+surprise+
65 sadness+anger+fear+disgust,data=data)
66 vif(fit2)
67 summary(fit2)
68
69 # (Tip Level) 1

```

The following object is masked from 'package:purrr':

```

some

```

	positive	negative	trust	joy	anticipation	surprise
> vif(fit1)	5.649534	1.826695	1.539096	2.555826	7.103988	2.800338
	sadness	anger	fear	disgust		
	4.229654	3.851841	2.901332	2.095943		

But I can believe on this only if that the particular these particular x variables are not correlated with each other. So to check that I will call for a library called car. So car library is not there we will quickly install this car library it is a small library it will not take much time. So car library helps you to check all this multi-collinearity and etcetera so that will get downloaded.

Okay it has been downloaded and if I now call the car library and want to check the multi-collinearity it will give me the variance inflation factors and you can see that anticipation has very high variance inflation factors 7.10 and sadness is also high. So our cut-off is generally 4 so I will first drop anticipation. So here I dropped anticipation and run this thing once more.

(Refer Slide Time: 32:11)

```
53 data=cbind(data,a)
54 names(data)[5:15]=c("positive","negative","trust","joy","anticipation","surprise")
55
56 fit=lm(Review_Overall_Rating~positive-negative,data=data)
57
58 fit2=lm(Review_Overall_Rating~positive-negative+trust+joy+anticipation+surprise+
59 sadness+anger+fear+disgust,data=data)
60
61 library(car)
62 vif(fit1)
63
64 fit2=lm(Review_Overall_Rating~positive-negative+trust+joy+surprise+
65 sadness+anger+fear+disgust,data=data)
66 vif(fit2)
67 summary(fit2)
68
```

```
positive negative trust joy surprise sadness anger fear disgust
3.800105 1.390139 1.460662 2.471649 2.375340 4.148129 3.138835 2.879490 2.084971
```

```
(intercept) 3.940386 0.064717 60.886 < 2e-16 ***
positive 0.034674 0.007558 4.587 5.10e-06 ***
negative 0.019172 0.010991 1.744 0.0814 .
trust 0.052018 0.010483 4.962 8.28e-07 ***
joy 0.007115 0.008723 0.816 0.4149
surprise -0.080289 0.017766 -4.519 7.00e-06 ***
anger -0.013737 0.021656 -0.634 0.5260
fear -0.068437 0.030745 -2.226 0.0262 *
disgust -0.238167 0.032711 -7.281 7.04e-13 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And if I now check it with the rest, the VIF is more or less okay still sadness is high probably. So I will also remove sadness from my model and then I will run the fit 2. So if that is the case then now after removing sadness and etcetera everybody VIF is smaller than 4 so no multi-collinearity anymore and this is my observation. See now I can tell that positive is properly contributing towards my overall rating now it makes sense.

Then trust is also significantly contributing towards my overall rating. Surprise has a negative so not everybody like surprise element do not keep even it is pleasant surprise preferably do not keep surprise elements. Fear and disgust probably disgust is the most contributing emotions towards the negative ratings. So these are some of the insights that you can generate

from the text data which can be used to call ultimately predict the quantitative data that here available.

So basic job is to find out from the text the emotions, the sentiments populated in a tabular form and then use the table for any prediction purpose. So that is where we will stop today. In the next video we will discuss a little bit about topic modeling. Thank you.