

Marketing Analytics
Prof. Swagato Chatterjee
Vinod Gupta School of Management
Indian Institute of Technology, Kharagpur
Lecture 36 - Recommendation Engine and Retail Analytics (Contd.)

Hello everybody. Welcome to Marketing Analytics course. This is Dr. Swagato Chatterjee from VGSOM IIT, Kharagpur who is taking this course. We are in week 7, session 3 and we will discuss about recommendation engine. In this particular video, I will do a hands-on on item to item collaborative filtering.

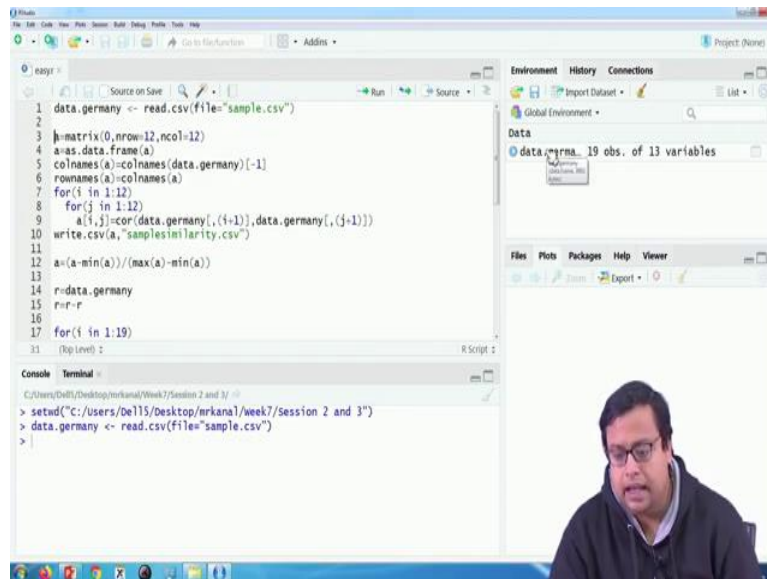
(Refer Slide Time: 0:36)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	user	a perfect cabba	ac/dc	adam gree	aerosmith	afi	air	alanis mor	alexisonfir	alicia keys	all that rer	amon	ama	macc amy
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	33	0	0	0	1	0	0	0	0	0	0	0	0	0
4	42	0	0	0	0	0	0	0	0	0	0	0	0	0
5	51	0	0	0	0	0	0	0	0	0	0	0	0	0
6	62	0	0	0	0	0	0	0	0	0	0	0	0	0
7	75	0	0	0	0	0	0	0	0	1	0	0	0	0
8	130	0	0	0	0	0	0	0	0	0	0	0	0	0
9	141	0	0	0	0	0	0	0	0	0	0	1	0	0
10	144	0	0	0	0	0	0	0	0	0	0	0	0	0
11	150	0	0	0	0	0	0	0	0	0	0	0	0	0
12	205	0	0	0	0	0	0	0	0	0	0	0	0	0
13	247	0	0	0	0	0	0	0	0	0	0	0	0	0
14	256	0	0	0	0	0	0	0	0	0	0	0	0	0
15	299	0	0	0	0	0	0	0	0	0	0	0	0	0
16	313	0	0	0	0	0	0	0	0	0	0	0	0	0
17	319	0	0	0	0	0	0	0	0	0	0	0	0	0
18	336	0	0	0	0	0	1	0	0	0	0	0	0	0
19	367	0	0	0	0	0	0	0	0	0	0	0	0	0

```

1 data.germany <- read.csv(file="sample.csv")
2
3 a=matrix(0,nrow=12,ncol=12)
4 a=as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
10 write.csv(a,"samplesimilarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r=r-r
16
17 for(i in 1:19)

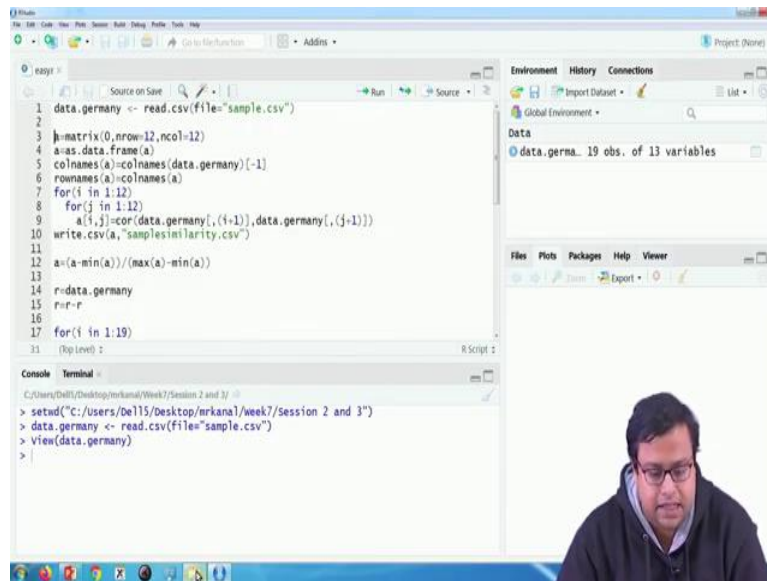
```



So to do that, there is this easy.r file that I have opened. Easy.r because there is another file which has been taken from Internet and that coding was very difficult to understand for a new comer. So I have changed that particular coding to a easier form and the data is about, so the data looks like this. So if I just open the data so this is a user, if you see it carefully, these are users and these are various I would say FM radio channels or radio songs and 0 and 1 means whether one user has listened to one particular thing or not. So this is something that you have to create.

The user in the left side. The items in the column and 0 and 1 is saying that whether something is seen or not, seen by a group of users. So based on that we are creating this and then the first thing is to read the data. So we will save working data into source file location and I will read the data. So the data this one is the smaller version of that data.

(Refer Slide Time: 1:59)



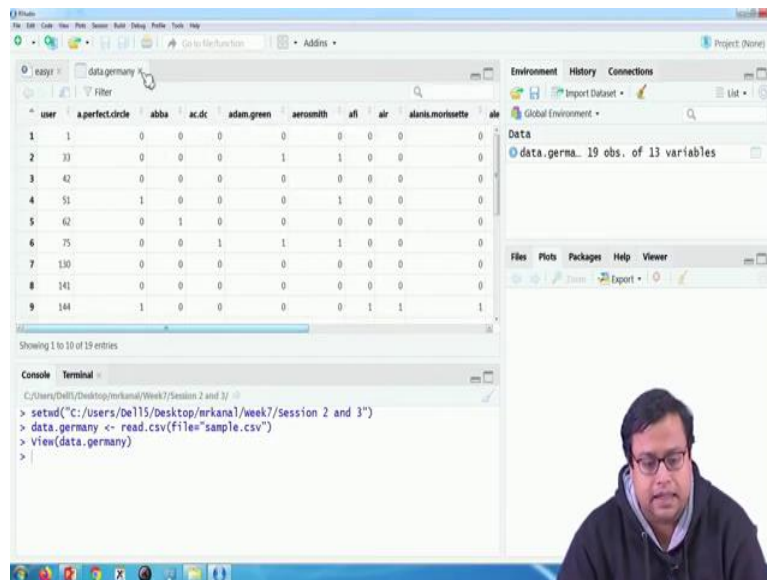
The screenshot shows the RStudio interface with the following R code in the editor:

```
1 data.germany <- read.csv(file="sample.csv")
2
3 h=matrix(0,nrow=12,ncol=12)
4 a=as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[, (i+1)],data.germany[, (j+1)])
10 write.csv(a,"sampleSimilarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r=r-r
16
17 for(i in 1:19)
18   (Top Level) :
```

The console shows the following commands and output:

```
> setwd("C:/Users/bell15/Desktop/mrkana1/week7/Session 2 and 3")
> data.germany <- read.csv(file="sample.csv")
> View(data.germany)
>
```

The Environment pane shows a data object named 'data.germany' with 19 observations and 13 variables.



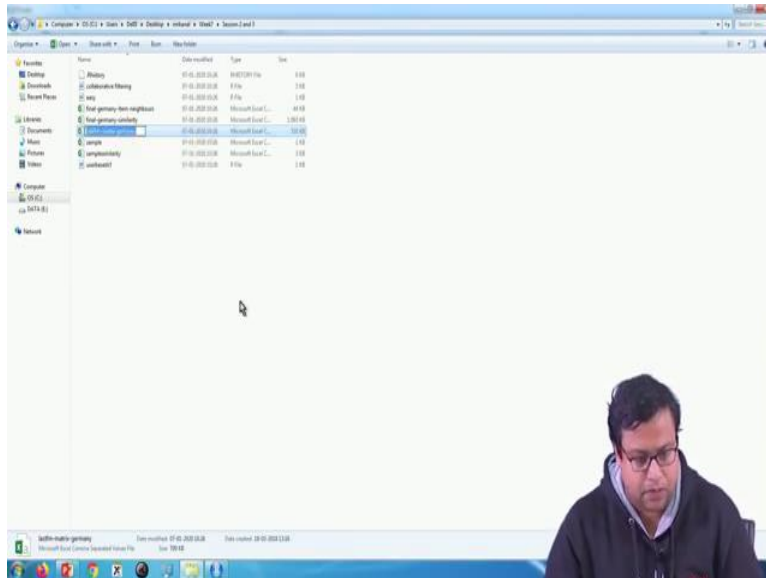
The screenshot shows the RStudio interface with the 'data.germany' object displayed in a data frame view. The data is as follows:

	user	a.perfectcircle	abba	ac.dc	adam.green	aeromith	afi	alr	alanis.morissette	ale
1	1	0	0	0	0	0	0	0	0	0
2	33	0	0	0	0	1	1	0	0	0
3	42	0	0	0	0	0	0	0	0	0
4	51	1	0	0	0	0	1	0	0	0
5	62	0	1	0	0	0	0	0	0	0
6	75	0	0	1	1	1	1	0	0	0
7	130	0	0	0	0	0	0	0	0	0
8	141	0	0	0	0	0	0	0	0	0
9	184	1	0	0	0	0	1	1	1	1

The console shows the following commands and output:

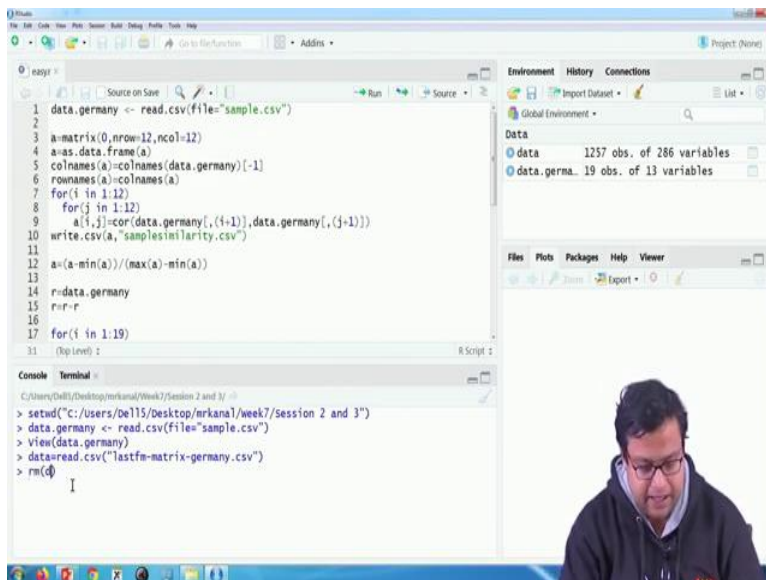
```
> setwd("C:/Users/bell15/Desktop/mrkana1/week7/Session 2 and 3")
> data.germany <- read.csv(file="sample.csv")
> View(data.germany)
>
```

The Environment pane shows a data object named 'data.germany' with 19 observations and 13 variables.



So I have taken only 19 users and only 13 variables but this is based on this sample.csv file which is the small file, 1 kb file but you can also do it with the bigger file. So you can change it to let us say last FM matrix Germany and you will get the bigger file and you can do all your calculations based on that. But because I have a time issue, it takes probably 3-4 minutes, 5 minutes in the calculations if I do it in the bigger file. So, if I do it with bigger file, it will have so just to give an example, so last FM Germany rename, copy.

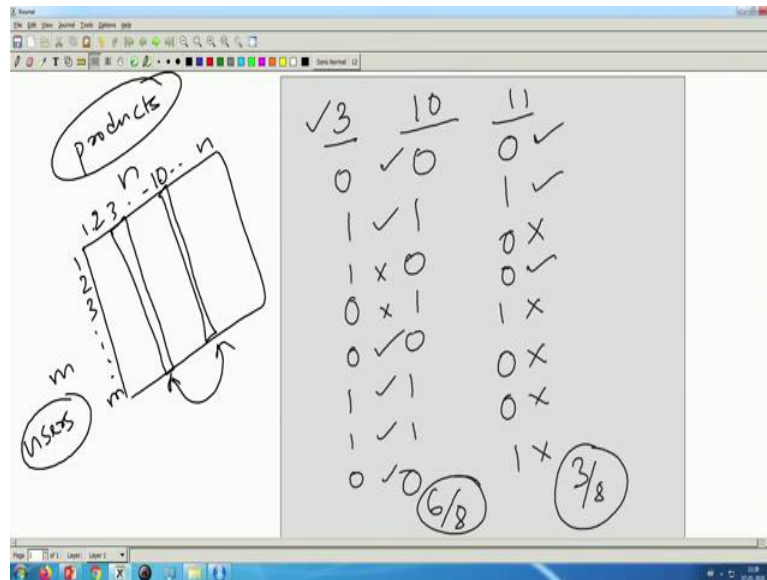
(Refer Slide Time: 2:44)



So if I just read that, you see there are 1,257 observations of 286 variables. 286 people are there and 1,257 observations are there. The calculation will take some 1 minute, 2 minute and I cannot sit idle in a video for 1 minute, 2 minute. So that is why I am using a small this thing, you can use the bigger one as well if you like. So, to given that average, this data 19

observations of 13 variables, I will start with this. So the first thing that we will do in the various steps of this calculation, we will create holder. So let us draw it.

(Refer Slide Time: 3:40)



So what is a holder? The first thing is see, here we have m number of customers and n number of products. This is what we have. So product 1, 2, 3 upto upto capital N and customer 1, 2, 3 up to small n. These are my users and these are my products. So from item to item, what do we do? I find out this item and let us say this item. What is the similarity between these two items? Let us say 3 and 10. The similarity is calculated by this way.

Let us say the item 3 is professor X and item 10 is professor Y and these are the students and the students has rated professor X as or in this case purchased so let us say student, various student taken the courses of professor X and professor Y. This guy has taken 1100110. Let us say this is the case and in professor 10, the guy has taken 01010110 and in professor 11 that is the another guy which has been taken 01001001. This is the thing that he has taken. Can you tell me by seeing that that which professor is closer to which professor?

Who, which two professors are more closer? You will see that these 2 professors whoever taken professor 3's class, how many times they have also taken professor 10's class? So professor 3's class and professor 10's class, this is a tick. This is a tick. This is a cross.

This is a cross. This is matching matching, matching, matching. So professor 3 and professor 10 out of 1, 2, 3, 4, 5, 6, 7, 8, 8 guys, 6 by 8 times the preferences matches. Whoever seen 3 has also taken classes of 10. Whoever has not taken classes of 3 have also not taken the classes of 10. That kind of matching has happened 6 times. But here this is match. This is

match. This is not a match. This is match. This is not a match. This is not a match. This is not a match. This is not a match. So 3/8 times. And that gives me an idea that 3 and 11 are closer than 3 and 10. So that is how we will be going to create a similarity matrix.

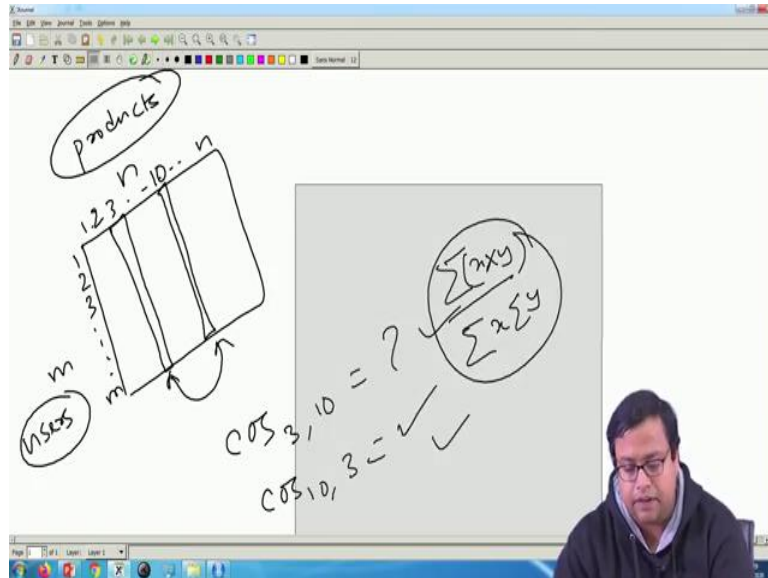
(Refer Slide Time: 6:25)

The diagram shows a square matrix representing a similarity matrix. The top row is labeled 'products' and contains indices 1, 2, 3, ..., 10, ..., n. The left column is labeled 'users' and contains indices 1, 2, 3, ..., m. A curved arrow points from the intersection of row 3 and column 10 to the text $\cos 3, 10 =$.

The video feed shows a man with glasses and a dark jacket, looking down at the screen.

```
17 for(i in 1:19)
18   for(j in 1:12)
19     if(data.germany[i, (j-1)] != 1){
20       k=sort(a[,j],decreasing=TRUE)[2:6]
21       l=order(-a[,j])[2:6]
22       h=as.numeric(data.germany[i, (1+1)])
23       r[i, (j+1)] = sum(h[k])/sum(l)
24     }
25
26
27
28
29 r$user=data.germany$user
30 reco=matrix(0,nrow=19,ncol=5)
31 reco=as.data.frame(reco)
32 rownames(reco)=r$user
33 colnames(reco)=c("Reco1", "Reco2", "Reco3", "Reco4", "Reco5")
```

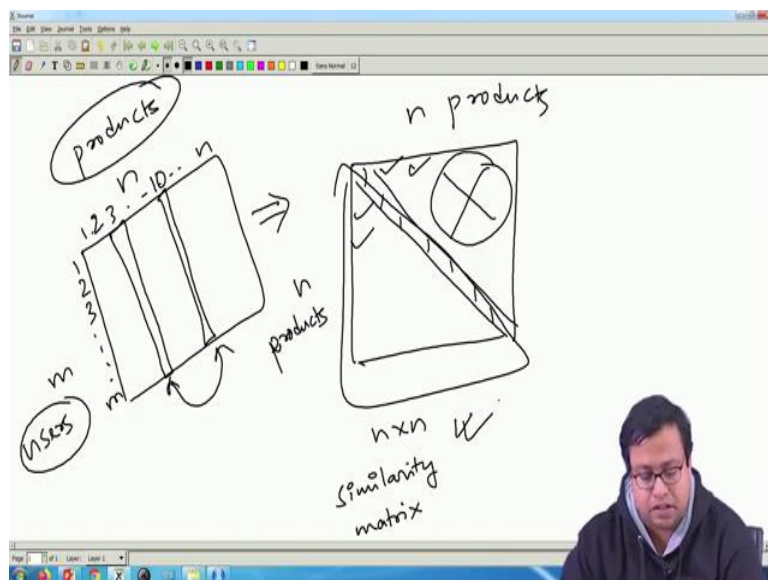
The video feed shows the same man from the previous slide, looking at the code.

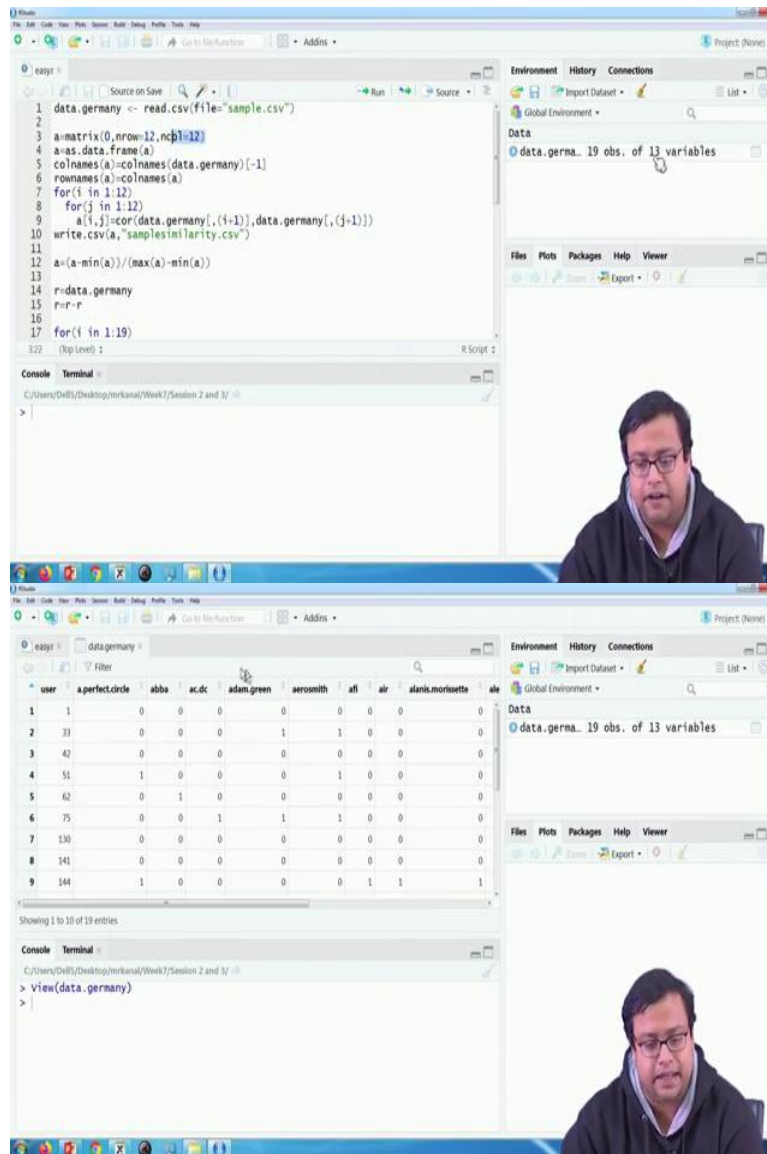


Now I will use the cosine matrix to do my calculation. So cosine is $\text{Cos}_{3,10}$ is basically the formula is $\frac{\sum(x y)}{\sqrt{\sum x^2 \sum y^2}}$. So this is the formula that has been written. I will come to that. Just one minute, $\frac{\sum(h k)}{\sqrt{\sum h^2 \sum k^2}}$. So that is what we are trying to use. So cosine just 1 minute. Cosine is yeah, so that is what I am trying to create as my cosine matrix and then we will try to find out. If I have to find out the similarity of $\text{Cos}_{3,10}$, what is the $\text{Cos}_{10,3}$?

Same thing, whatever I get here. So the formula is basically this matrix \times this matrix by, $\frac{\sum(x \times y)}{\sqrt{\sum x^2 \sum y^2}}$. If that is my cosine then I will get the similar thing here as well.

(Refer Slide Time: 7:41)





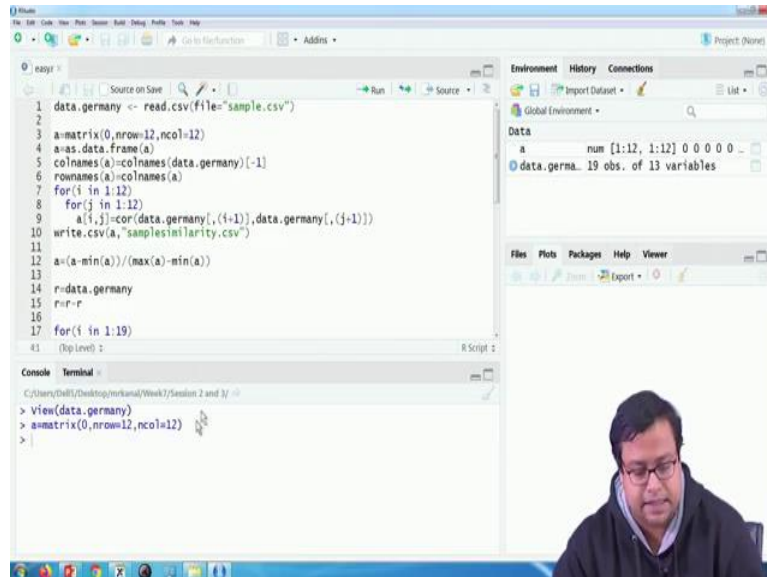
So if I get the similar thing here as well then the if I try to plot them in a, if I try to convert it into a matrix, how will it look like? Each item to each item, so this will create n number of products here and n number of products here and this will be a similarity matrix. So n into n similarity matrix gets created. Fair enough. Now what is the similarity of the diagonal elements? It will be exactly 1. Because I am absolutely fully similar to me.

What and then it will be a either upper triangular or lower triangular symmetric. So whatever is value here, comes here. Whatever value here, comes here and so on. So upper side and lower side is symmetric. So one set of calculation you do, other set of calculation you do not do, it is okay.

So that is how we are reducing the number of calculations. So this is what I will first try. And find out how items are closer to another item. So to create this matrix, to populate this matrix, I have to create the matrix holder. We are naming it as A. A is a matrix where all the entries

are 0. Number of row is 12. Number of column is 12. Why? Because there are 13 variables. Now out of these 13 variables the first variable is user name, so that is why we are not using. The rest there are 12 items, that is why 12 by 12 matrix I am creating.

(Refer Slide Time: 9:20)



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

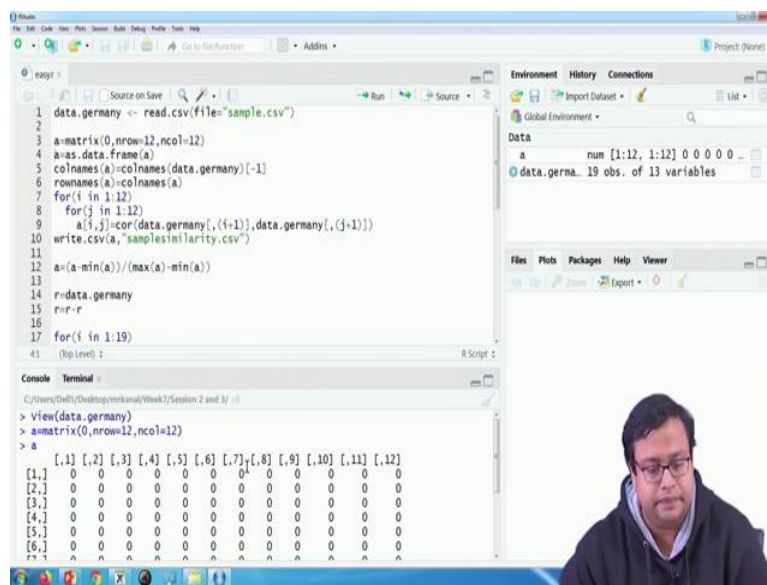
```
1 data.germany <- read.csv(file="sample.csv")
2
3 a=matrix(0,nrow=12,ncol=12)
4 a=as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[, (i+1)],data.germany[, (j+1)])
10 write.csv(a,"sampleSimilarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r=r-r
16
17 for(i in 1:19)
41 (Dip Level) :
```

The Environment pane on the right shows the following objects:

- Global Environment
- Data
 - a num [1:12, 1:12] 0 0 0 0 0 ...
 - data.germany 19 obs. of 13 variables

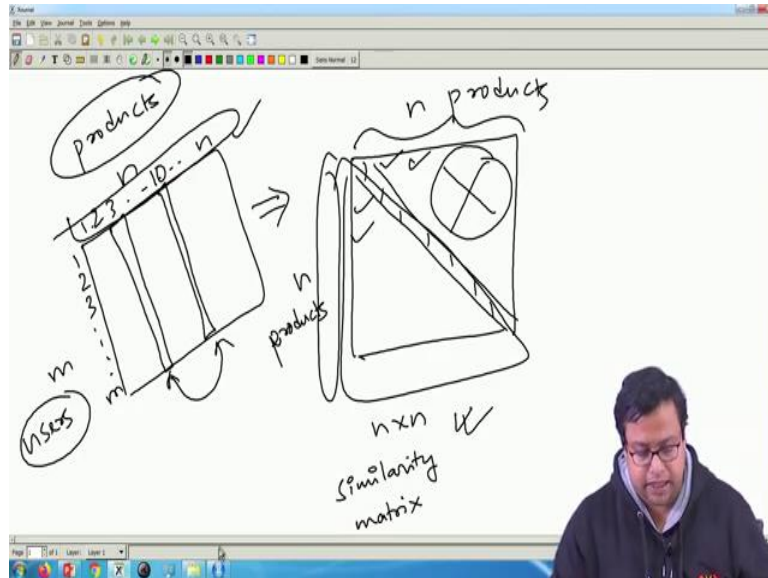
The Console shows the following output:

```
> view(data.germany)
> a=matrix(0,nrow=12,ncol=12)
>
```



The screenshot shows the RStudio interface. The script editor contains the same R code as the previous screenshot. The Environment pane on the right shows the same objects as the previous screenshot. The Console shows the following output:

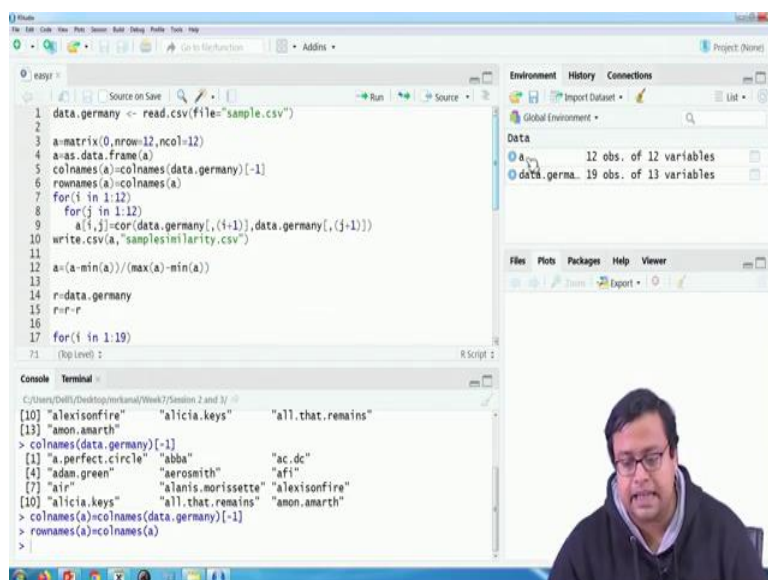
```
> view(data.germany)
> a=matrix(0,nrow=12,ncol=12)
> a
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
[1,] 0 0 0 0 0 0 0 0 0 0 0 0
[2,] 0 0 0 0 0 0 0 0 0 0 0 0
[3,] 0 0 0 0 0 0 0 0 0 0 0 0
[4,] 0 0 0 0 0 0 0 0 0 0 0 0
[5,] 0 0 0 0 0 0 0 0 0 0 0 0
[6,] 0 0 0 0 0 0 0 0 0 0 0 0
[7,] 0 0 0 0 0 0 0 0 0 0 0 0
```



Right now this A matrix is a blank matrix, it is a blank matrix nothing is written here. Fair enough. Now I change to, create change into a data frame and then what will be the name of the data frame? Remember this is the matrix that looks like this. The row names will be the row names first of all, the column names of this matrix will be similar to these values, the column names of this matrix and the row names will also be the same.

So the row names and column names both will be similar to the column names of the first original raw dataset.

(Refer Slide Time: 10:00)



```

1 data.germany <- read.csv(file="sample.csv")
2
3 a=matrix(0,nrow=12,ncol=12)
4 a=as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
10 write.csv(a,"sample1similarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r=r-r
16
17 for(i in 1:19)

```

```

> a=as.data.frame(a)
> colnames(data.germany)
[1] "a.perfect.circle" "abba"
[4] "ac.dc"            "adam.green"      "aerosmith"
[7] "afi"             "alicia.keys"     "alanis.morissette"
[10] "alexisonfire"   "all.that.remains"
[13] "amon.amarth"
>

```

	a.perfect.circle	abba	ac.dc	adam.green	aerosmith	afi	air	alanis.morissette
a.perfect.circle	0	0	0	0	0	0	0	0
abba	0	0	0	0	0	0	0	0
ac.dc	0	0	0	0	0	0	0	0
adam.green	0	0	0	0	0	0	0	0
aerosmith	0	0	0	0	0	0	0	0
afi	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0
alanis.morissette	0	0	0	0	0	0	0	0
alexisonfire	0	0	0	0	0	0	0	0

```

[13] "amon.amarth"
> colnames(data.germany)[-1]
[1] "a.perfect.circle" "abba"          "ac.dc"
[4] "adam.green"      "aerosmith"    "afi"
[7] "aia"            "alanis.morissette" "alexisonfire"
[10] "alicia.keys"    "all.that.remains" "amon.amarth"
> colnames(a)=colnames(data.germany)[-1]
> rownames(a)=colnames(a)
> view(a)
>

```

So column names of A = colnames(data.Germany)-1? Because first entry is user that I will not take. So if I just take this part, copy it and run it here, it gives me 13 entries with the first name as user. So this one I will not take. This one I will not take. So that is why I am writing this and I am not getting the user 1, the rest 12 is I am getting.

I am putting them as column names of A and I am also saying that row names of A and column names of A is same. That means the row names and column names are same. Now how does A looks like at this moment? A looks like this. All the entries are 0, we will do the calculations but these are the product names, these are some product names, fair enough.

(Refer Slide Time: 10:55)

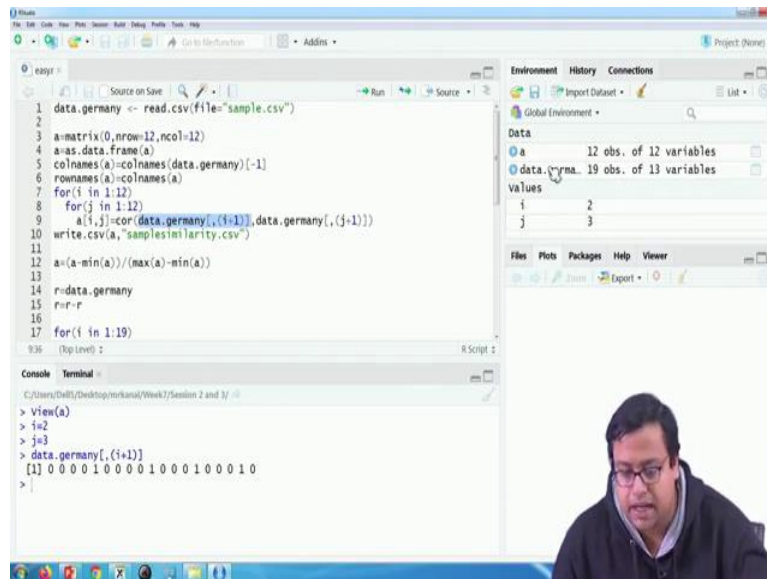
The screenshot shows the RStudio interface. The main editor displays a data frame 'a' with 12 rows and 8 columns. The rows are named with names like 'a.perfect.circle', 'abba', 'ac.dc', 'adam.green', 'aeromith', 'afi', 'air', 'alanik.morissette', and 'alexionfire'. The columns are numeric. The console shows the command 'View(a)'.

	a.perfect.circle	abba	ac.dc	adam.green	aeromith	afi	air	alanik.morissette
a.perfect.circle	0	0	0	0	0	0	0	0
abba	0	0	0	0	0	0	0	0
ac.dc	0	0	0	0	0	0	0	0
adam.green	0	0	0	0	0	0	0	0
aeromith	0	0	0	0	0	0	0	0
afi	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0
alanik.morissette	0	0	0	0	0	0	0	0
alexionfire	0	0	0	0	0	0	0	0

```
> View(a)
```

The screenshot shows the RStudio interface with R code in the editor. The code reads a CSV file, creates a matrix, calculates a correlation matrix, and normalizes it. The console shows the execution of the code.

```
1 data.germany <- read.csv(file="sample.csv")
2
3 a=matrix(0,nrow=12,ncol=12)
4 a=as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
10 write.csv(a,"samplesimilarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 P=r-r
16
17 for(i in 1:19)
18   (i)
19
```



Now what I will do is I will calculate the cosine or here I have calculated the correlation. So correlation is also fine, I think. So how to calculate the correlation? So easy formula is correlation, what is about, so higher the correlation the higher is my I would say the higher is the similarity. So for (i in 1:12), for (j in 1:12) so this is something for (i in 1:12) for (j in 1:12), $a[i, j]=\text{col}(\text{data.germany}[(i+1)], \text{data.germany}[(j+1)])$. Why + 1?

Because the first column is the user column. The first column is user column. So let us say to give an idea, I want to populate in this a I want to populate this value, let us say this value which is ac.dc and abba. In the column it is third column, in the row it is the second row. So what is the case? The third column second row. So $i = 2, j = 3$, fair enough. Then what is basically correlation $i + 1$? So what is this value? This value is 0 0 0 1 0 0 0 0 1 0 0 0 1 something like that which is basically the third column.

(Refer Slide Time: 12:36)

The screenshot shows the RStudio interface. The Environment pane on the right displays a data frame with 19 observations and 13 variables. The console shows the following R code and output:

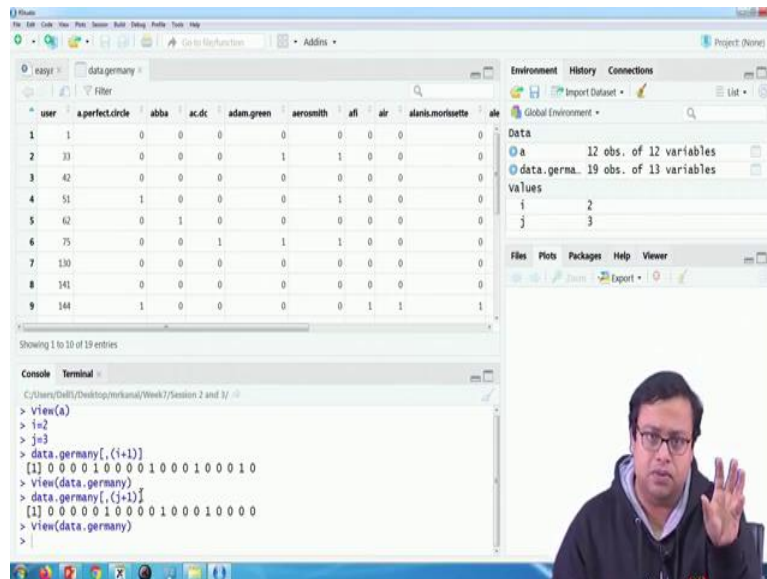
```
> View(a)
> i=2
> j=3
> data.germany[(i+1)]
[1] 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0
> View(data.germany)
```

user	aperfectcircle	abba	ac.dc	adam.green	aeromith	afi	air	alanismorissette	ale
7	130	0	0	0	0	0	0	0	0
8	141	0	0	0	0	0	0	0	0
9	144	1	0	0	0	0	1	1	1
10	150	0	1	0	0	0	0	0	0
11	205	0	0	1	0	0	0	0	0
12	247	0	0	0	1	0	0	0	0
13	256	0	0	0	0	0	0	0	1
14	299	1	1	0	0	0	0	0	0
15	313	0	0	1	0	0	0	0	0

The screenshot shows the RStudio interface with R code being executed. The console shows the following R code and output:

```
1 data.germany <- read.csv(file="sample.csv")
2
3 a=matrix(0,nrow=12,ncol=12)
4 a=as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
10 write.csv(a,"sampleSimilarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r=r-r
16
17 for(i in 1:19)
18   for(j in 1:19)
19     r[i,j]=a[i,j]+r[i,j]
```

```
> View(a)
> i=2
> j=3
> data.germany[(i+1)]
[1] 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0
> View(data.germany)
> data.germany[(j+1)]
[1] 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0
```



If you check carefully this is the third column, 1, 2, 3 this column 0 0 0 1 0 0 0 1 and then 0 0 0 1 and 0 this is what I am populating here. So this is the third column and with that $j = 3$ that means the fourth column I will take a combination. The fourth column looks like this. What is the fourth column?

In the data Germany the fourth column is acdc. So these two columns correlation I will try to find out. That is why $i + 1, j + 1$ and then after putting finding out, I will put them into my matrices or a datasets i, j th cell I will populate the value. So that is what I am trying to do and actually I should not have calculated 1 to 12. It would have been better if I have calculated from $i = 1$ to i rather than 1 to 12, $j = 1$ to i because see it is a matrix which is lower triangular matrix or upper triangular matrix.

Other things are same, it is a symmetric. So I could have calculated lower triangular matrix and find out the symmetry but I have not done that because it is a smaller size, it will not take time so that is why I have done this. So $[i \ a][i \ j] = \text{correlation of this and this}$.

(Refer Slide Time: 14:10)

The RStudio interface displays a data frame with 12 rows and 12 columns. The columns are named: a.perfect.circle, abba, ac.dc, adam.green, aerosmith, afi, air, and alank.mor. The values are numerical correlation coefficients. The console window shows the following R code:

```
> for(i in 1:12)
+ for(j in 1:12)
+ a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
> view(a)
```

The RStudio interface shows a script with the following R code:

```
1 data.germany <- read.csv(file="sample.csv")
2
3 a=matrix(0,nrow=12,ncol=12)
4 as.data.frame(a)
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8 for(j in 1:12)
9 a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
10 write.csv(a,"samplesimilarity.csv")
11
12 k=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r=r-r
16
17 for(i in 1:19)
18
19
```

The console window shows the following R code:

```
> for(i in 1:12)
+ for(j in 1:12)
+ a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
> view(a)
> write.csv(a,"samplesimilarity.csv")
```


The screenshot shows the RStudio interface. The main window displays a correlation matrix for variables: a.perfect.circle, abba, ac.dc, adam.green, aerosmith, afl, air, and alank.mor. The matrix is symmetric with 1s on the diagonal. The value for the correlation between 'a.perfect.circle' and 'abba' is 0.1304373. The console shows the following R code:

```

> for(i in 1:12)
+   for(j in 1:12)
+     a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
> view(a)
> write.csv(a,"samplesimilarity.csv")
> view(a)

```

So if I run three lines together, I get the all the correlation values. See, it is 0.13043, 13043 it is symmetric to over the diagonal. All the diagonal elements are 1 and the other correlation values are written here, fair enough. Now this is something that I save because this is something that I will use later.

If it is a huge dataset it is better to save it. Now what will I do next? I, you remember that these correlations are between correlation values. Correlation values are between - 1 to + 1. I am trying to make sure that all of them are scaled properly so that remain from 0 to 1. So what do I am writing? I am writing it is called normalization. We have done this before.

(Refer Slide Time: 15:09)

The whiteboard shows the following handwritten text and formulas:

old value = x

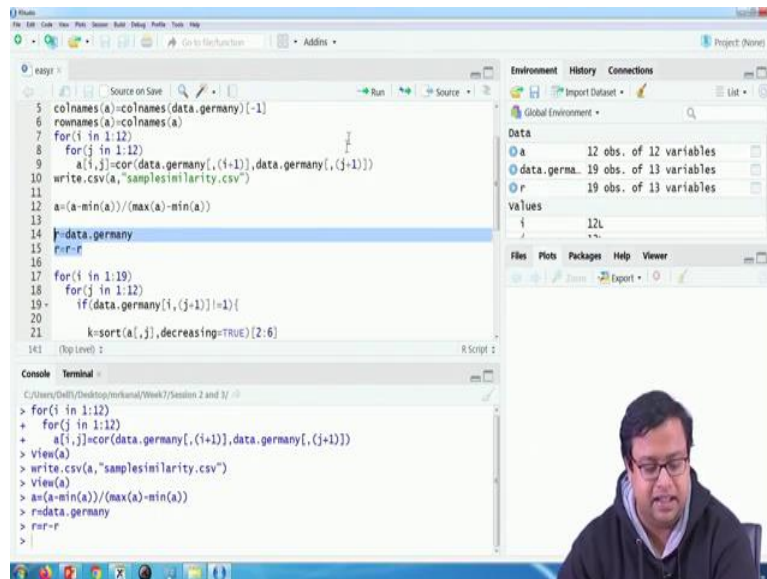
$$\text{new value} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$= \frac{x - (-0.73)}{1 - (-0.73)}$$

$$= \frac{x + 0.73}{1.73}$$

min(x) = -0.73
max(x) = 1

$x \rightarrow 0$
 $n.v. = \frac{0.73}{1.73}$ ✓



The normalization is done in this way. So I am saying that the new item new value $a=(a-\min(a))/(\max(a)-\min(a))$. Let us say the minimum of x is -0.73 and maximum of x is all I know that 1 because 1 we are getting. So then this value will be $x - (-0.73) / 1 - (-0.73)$ or in other words, $x + 0.73 / 1.73$, fair enough. This is the value that I get.

Now let us say one particular value is 1. When it is 1, the new value will be $1 + 0.73 / 1 + 0.73$, it will be 1. If the value is -0.73 then the numerator becomes 0. This value becomes 0 and the denominator stays. So then I know that okay numerator becomes 0, denominator stays so that means that the minimum value is 0. If x value is 0 something in between then the value will be when $x = 0$, this new value will be $0.73 / 1.73$ or something like that whatever value comes here.

So it is now covered between 0 and 1, so that is what I am trying to do here. I am putting them into 0 and 1, fair enough. I am putting them between 0 and 1. Now what next? Now I am trying to create a matrix which is a similarity matrix which is a recommendation matrix which called r . So I wrote $r = \text{data.germany}$ then $r = (r-r)$. By running these two lines simply what I created is 19, these are my users, these are my items. Okay, sorry, this is something that I probably have done wrong.

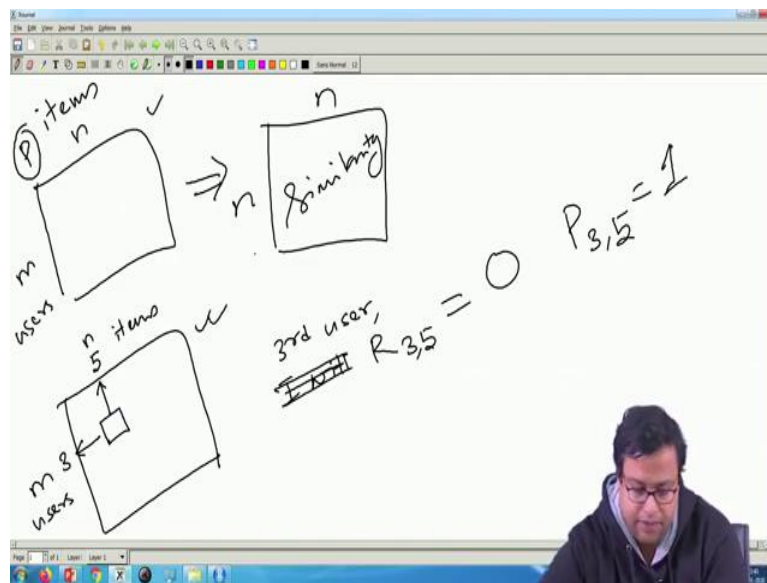
(Refer Slide Time: 17:28)

```
5 colnames(a)=colnames(data.germany)[-1]
6 rownames(a)=colnames(a)
7 for(i in 1:12)
8   for(j in 1:12)
9     a[i,j]=cor(data.germany[(i+1)],data.germany[(j+1)])
10 write.csv(a, "samplesimilarity.csv")
11
12 a=(a-min(a))/(max(a)-min(a))
13
14 r=data.germany
15 r[2:13]=0
16
17 for(i in 1:19)
18   for(j in 1:12)
19     if(data.germany[i,(j+1)]!=1){
20       k=sort(a[,j],decreasing=TRUE)[2:6]
21     }
```

user	a:perfectcircle	abba	ac.dc	adam.green	aerosmith	afi	alr	alanis.morissette	ale
1	1	0	0	0	0	0	0	0	0
2	33	0	0	0	0	0	0	0	0
3	42	0	0	0	0	0	0	0	0
4	51	0	0	0	0	0	0	0	0
5	62	0	0	0	0	0	0	0	0
6	75	0	0	0	0	0	0	0	0
7	130	0	0	0	0	0	0	0	0
8	141	0	0	0	0	0	0	0	0
9	144	0	0	0	0	0	0	0	0

r = data.germany and then I would write r sorry, 13 variables. So 2 to 13 = 0 that is the right thing to do. I am sorry. So if I run this then what do I get? The users, the items and currently all these items are 0. So I will find out the recommendation scores for these items. How do I do it? Now what did I do till now? I created only a similarity matrix.

(Refer Slide Time: 18:16)



```
13 r=data.germany
14 r[2:13]=0
15
16
17 for(i in 1:19)
18   for(j in 1:12)
19     if(data.germany[i,(j+1)]!=1){
20
21       k=sort(a[,j],decreasing=TRUE)[2:6]
22       l=order(-a[,j])[2:6]
23       h=as.numeric(data.germany[i,(1-1)])
24
25       r[i,(j+1)]=sum(h*k)/sum(k)
26
27
28 r$user=data.germany$user
29
266 (Dip Level) :
```

The screenshot shows the RStudio environment with the following code in the script editor:

```
13 r=data.germany
14 r[2:13]=0
15
16
17 for(i in 1:19)
18   for(j in 1:12)
19     if(data.germany[i,(j+1)]!=1){
20
21       k=sort(a[,j],decreasing=TRUE)[2:6]
22       l=order(-a[,j])[2:6]
23       h=as.numeric(data.germany[i,(1-1)])
24
25       r[i,(j+1)]=sum(h*k)/sum(k)
26
27
28 r$user=data.germany$user
29
266 (Dip Level) :
```

The console shows the following output:

```
> write.csv(a,"samplesimilarity.csv")
> view(a)
> as(a-min(a))/(max(a)-min(a))
> r=data.germany
> rer=r
> view(r)
> r=data.germany
> r[2:13]=0
> view(r)
```

If you remember I have created a similarity matrix which is saying that the higher the value the more is the similarity, the lower the value the less is the similarity. So I started with m users, n items, from here I have created n by n similarity matrix. This is what I have created.

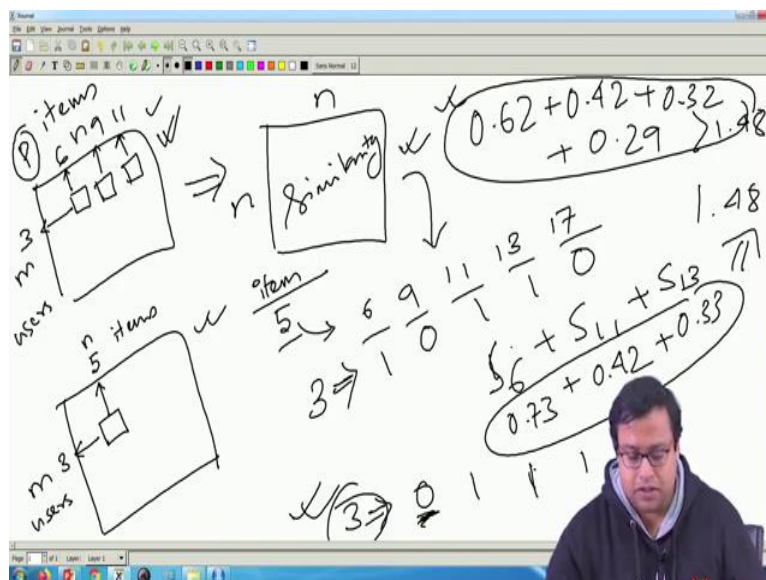
Now I have created recommendation matrix which is currently blank which has m users and n items. So now choose any one cell. Let us say for this cell the user number is 3, the item number is 5. Now for the third user I will, my recommendation decision, the recommendation decision $R_{3,5}$, the recommendation decision will always be 0 irrespective of anything.

If this is the purchase history, the purchase 3, 5 is 1. That means if I have already purchased it, I will never recommend. So the first thing that I will check is that whether this value is 1 in this case or not. If it is 1 here it will always be 0 here. So that is what I have written here.

Carefully you see that for $i = 1$ to 19 and for $j = 1$ to 12, 1 to 19 means the number of observation, number of customers and $j = 1$ to 12, this happens only if data.germany the original purchase data in that the value is 0. If it is not = 1 then only happens, otherwise it does not happen, otherwise this calculation will not happen. The value will remain 0. The value in the recommendation will remain 0. If by chance this is 1, this \neq I have written, by chance this value is = 1, that means in the purchase data third customer has seen fifth product then I will not recommend that product anymore.

If third customer has not seen the fifth product then I may recommend based on the calculations that we are going to do but if he has already seen, I will never recommend. So that is why this thing. Now if he has not seen then what is the question? Then what? If by chance let us say this guy has not seen it, so then if he has not seen then what do I do?

(Refer Slide Time: 21:15)



I simply find out that, okay, item number m if this is the item, which are the top similar items? So let us say I find out item number m has similar items from this matrix, I will get actually this information. That item number 6, 9, 11, 13 and 17, these five items are most similar items to item number 5. Now I will recommend item number 5 to user 3. Listen to this carefully. I will recommend Main hoon na to my friend.

Here we are doing item to item collaborative filtering. So all the similarity is measured based on items. So I will recommend Main hoon na to one of my friend only if I have seen that my friend has also seen similar movies like Main hoon na. Listen to this carefully once more. I will recommend one movie, Main hoon na to my friend or anybody if I know that that somebody has also seen movies which are similar to Main hoon na, fair enough. So which are

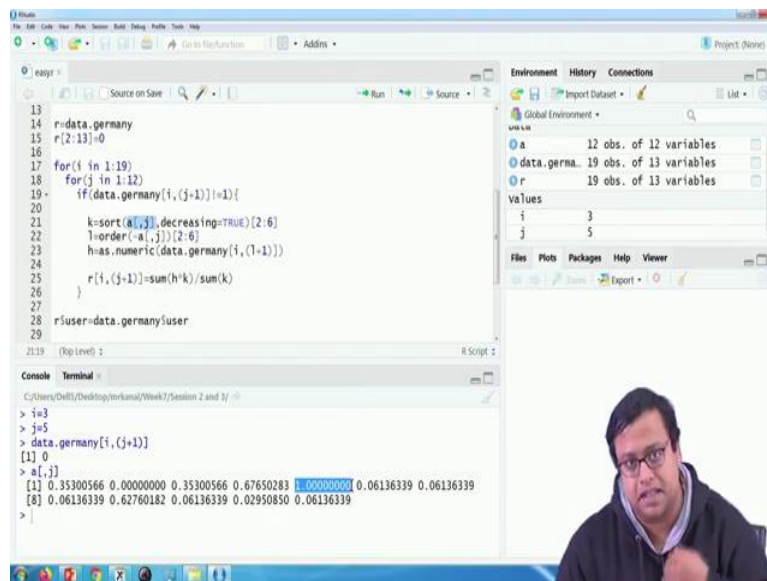
similar to Main hoon na? If Main hoon na is item 5 then item 6, 9, 11, 13 and 17 are similar to Main hoon na.

Now for this 6, 9, 11, 13 and 17 whether they have seen it, from which matrix I will get? From this matrix that whether user number 3 has seen it. So user number 3 has seen the sixth item will come here, this cell where this is 6, this is 3; has seen the ninth item, this is 9, this is 3, I have seen the eleventh item, this is 11, this is 3. That is how I will find out whether this guy has seen these items and I see that okay, this guy has seen this, not seen this, this, this and this. Fair enough.

Then his net similarity score is basically 1, basically similarity of 6 + similarity of 11 + similarity of 13. Let us say 6 similarity is 0.73 means similarity of item number 6 with item number 5 is 0.73, item number 11 with item number 5 is 0.42 and item number 13 to item number 5 is 0.33. So this is his net similarity score for product 3. On the other hand, let us say in another situation I see that this guy has seen not seen this this, this, this, this the rest four he has seen, okay and if the rest four he has seen, corresponding values are coming, let us say 0.62, 0.42, 0.32 and 0.29 in the second case.

This is in the second case. So you see in the first case the total value is $0.73 + 0.42$ that means $1.15 + 0.33$, 1.48 and in the second case it is coming more than 1.48. So in the second case the recommendation, chances of recommending item 5 will be much higher though he has not seen the most closest movie. So this is something that is what we are doing. So what I am doing carefully you see. What I do?

(Refer Slide Time: 25:08)



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

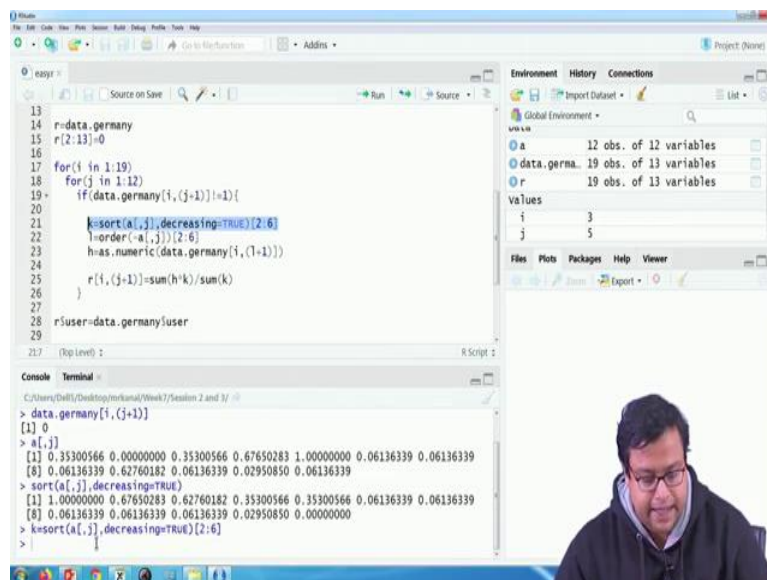
```
13
14 r=data.germany
15 r[2:13]=0
16
17 for(i in 1:19)
18   for(j in 1:12)
19     if(data.germany[i,(j-1)]!=1){
20
21       k=sort(a[,j],decreasing=TRUE)[2:6]
22       l=order(-a[,j])[2:6]
23       h=as.numeric(data.germany[i,(1-1)])
24
25       r[i,(j+1)]=sum(h*k)/sum(k)
26
27     }
28 r$user=data.germany$user
29
```

The console shows the execution of the loop for i=1 and j=5:

```
> i=1
> j=5
> data.germany[i,(j+1)]
[1] 0
> a[,j]
[1] 0.35300566 0.00000000 0.35300566 0.67650283 1.00000000 0.06136339 0.06136339
[8] 0.06136339 0.62760182 0.06136339 0.02950850 0.06136339
>
```

The Environment pane on the right shows the following objects:

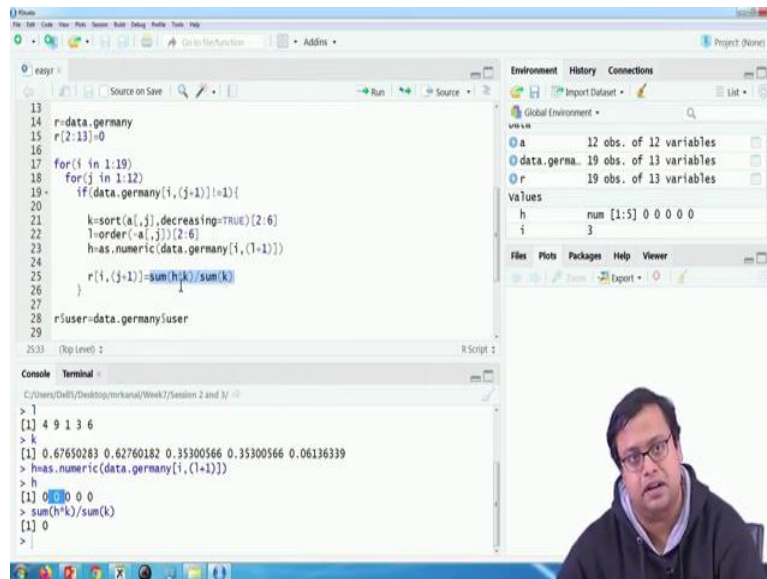
- Global Environment
 - a: 12 obs. of 12 variables
 - data.germany: 19 obs. of 13 variables
 - r: 19 obs. of 13 variables
- Values
 - i: 3
 - j: 5



The screenshot shows the RStudio interface with the same script as above. The line `k=sort(a[,j],decreasing=TRUE)[2:6]` is highlighted in blue. The console shows the execution of the sort function:

```
> data.germany[i,(j+1)]
[1] 0
> a[,j]
[1] 0.35300566 0.00000000 0.35300566 0.67650283 1.00000000 0.06136339 0.06136339
[8] 0.06136339 0.62760182 0.06136339 0.02950850 0.06136339
> sort(a[,j],decreasing=TRUE)
[1] 1.00000000 0.67650283 0.62760182 0.35300566 0.35300566 0.06136339 0.06136339
[8] 0.06136339 0.06136339 0.06136339 0.02950850 0.00000000
> k=sort(a[,j],decreasing=TRUE)[2:6]
```

The Environment pane on the right is identical to the previous screenshot.



I do $k = \text{sort}(a[,j], \text{decreasing} = \text{TRUE})[2:6]$. What is a comma j ? j is the product. So I will just quickly check this. Let us say $i = 3$ and $j = 5$, $\text{data.germany}[i,(j+1)]$ what is the value? 0, so I can go ahead. He has not seen before. Now what is this $a[,j]$? That means a that similarity matrix j -th column. These are the similarity of j -th product that means fifth product with other products. You see the fifth product similarity with fifth product is 1 this similarity with himself is 1 and these are the rest of the thing. Now if I try to find out which product is the most similar, I have to sort this out, fair enough.

So that is what I am doing, sorting. I am sorting it out and it comes to be the first item to be 1 because I am most similar to myself but it has no meaning. The next 5 items' similarity is like this, fair enough. So that is why I take from 2 to 6. So k is $= \text{sort}(a[,j][2:6])$. So k is basically the top 5 items, item number 5 whichever other items are closest to item number 5 corresponding similarity scores are this and if I want to find out order, order means the serial number. Serial number of $-a[,j]$ will give me the order.

Fifth item is most similar to fifth item then fourth item then ninth item then first item then third item and so on. So these are the order, this is not the similarity score. This is the order of the similarity score. So that 2 to 6 if I put it in 1, this gives my identity. So fourth, ninth, first third and sixth items are the top most similar items to item number 5 and corresponding similarity scores are these. So k stores the similarity scores, l stores the similar items.

Now I have to check out of these whatever items has been stored in l whichever has been seen by the i -th customer, in this case the second customer or third customer, $i = 3$, the third customer whether he has seen it or not. So how will I know? This $\text{data.germany}[i,(1+1)]$ because the first column in the data.germany dataset is the users. So this is my history.

The history is says h is 0 0 0 0 0. So he has not seen any of these movies. Then how much will be the probability? 0 because 0 is my history into the similarity score by the sum of the similarity score. So that is something that we calculate and put it in $r[i, (j + 1)]$, so the value becomes 0. This particular value will still become 0 because all my h items are 0. If by chance one of them were 1 any one of them were 1, corresponding value would have been calculated.

(Refer Slide Time: 28:49)

The screenshot shows the RStudio interface with the following code in the editor:

```

13
14 r=data.germany
15 r[,2:13]=0
16
17 for(i in 1:19)
18 for(j in 1:12)
19 if(data.germany[i,(j+1)]!=1){
20
21 k=sort(a[,j],decreasing=TRUE)[2:6]
22 l=order(-a[,j])[2:6]
23 h=as.numeric(data.germany[l,(1+1)])
24
25 r[i,(j+1)]=sum(h*k)/sum(k)
26
27
28 r$user=data.germany$user
29

```

The Environment pane shows the variable `r` with 19 observations and 13 variables. The values for `h`, `i`, `j`, `k`, and `l` are displayed.

The Console shows the execution of the code:

```

C:/Users/Dalit/Desktop/mkranal/Week7/Session 2 and 3/ >
+ for(j in 1:12)
+ if(data.germany[i,(j+1)]!=1){
+
+ k=sort(a[,j],decreasing=TRUE)[2:6]
+ l=order(-a[,j])[2:6]
+ h=as.numeric(data.germany[l,(1+1)])
+
+ r[i,(j+1)]=sum(h*k)/sum(k)
+
+ }
>

```

The screenshot shows the RStudio interface displaying the resulting similarity matrix `r`. The Environment pane shows the variable `r` with 19 observations and 13 variables. The values for `h`, `i`, `j`, `k`, and `l` are displayed.

The Console shows the execution of the code:

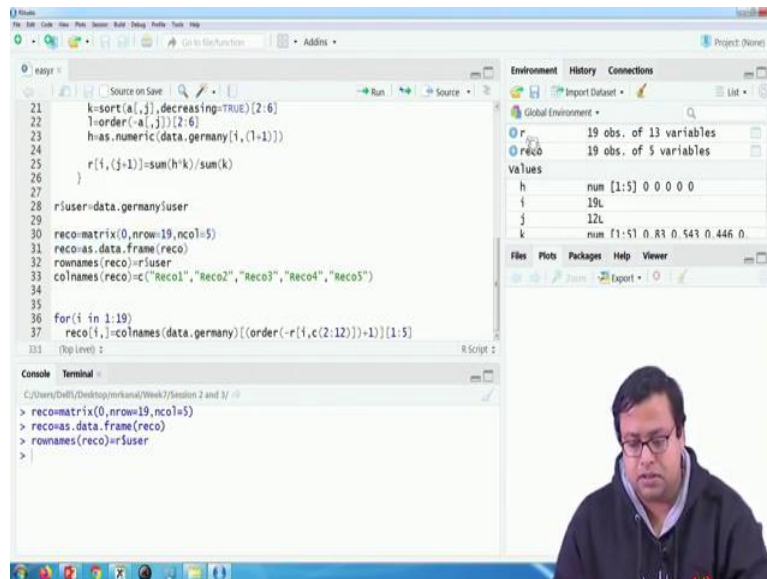
```

C:/Users/Dalit/Desktop/mkranal/Week7/Session 2 and 3/ >
+ if(data.germany[i,(j+1)]!=1){
+
+ k=sort(a[,j],decreasing=TRUE)[2:6]
+ l=order(-a[,j])[2:6]
+ h=as.numeric(data.germany[l,(1+1)])
+
+ r[i,(j+1)]=sum(h*k)/sum(k)
+
+ }
> view(r)
>

```

The resulting similarity matrix `r` is shown below:

	user	a.perfectcircle	abba	acdc	adam.green	aerosmith	all	air	alanic.morisset
1	1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	33	0.1623689	0.0000000	0.38355342	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
3	42	0.0000000	0.0000000	0.24253377	0.0000000	0.0000000	0.0000000	0.03950013	0.0000000
4	51	0.0000000	0.26748297	0.1917671	0.38009255	0.0000000	0.1767813	0.20331835	0.21
5	62	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
6	75	0.1623689	0.07697917	0.0000000	0.0000000	0.0000000	0.0000000	0.05321928	0.0000000
7	130	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
8	141	0.0000000	0.26748297	0.0000000	0.0000000	0.0000000	0.1767813	0.0000000	0.0000000
9	144	0.0000000	0.30260185	0.03333677	0.06895394	0.20005513	0.0000000	0.0000000	0.0000000



Now this is the operation that I do for all my products and all my users. If you did not understand this, please stop the video, go back and understand it properly. So I run that. Now I have created the r matrix, some of the values are coming 0. 0 can come for two reasons, one is this guy has already seen the movie before that is why it is coming 0 or this guy has seen no similar movie. We have taken the top five, no similar top five movies, that is why the value is coming 0.

These are the two reasons why these values can come 0. Whenever he has seen at least one similar movie then some value will come here. Now what I do is next, I will create the recommendation. So these are the recommendations course. I need actual names of the movies. So what I create is I create a recommendation matrix which has 19 rows and 5 columns.

Why 5 columns? Because top 5 recommendation will come. In normal case in Netflix and etcetera 5 recommendations come so top 5. The row names will be the user names. So that is how the user names are coming here.

(Refer Slide Time: 30:04)

The screenshot shows the RStudio interface. The main editor displays a data frame named 'reco' with 19 rows and 5 columns (V1-V5). The values are all 0. The console shows the following R code:

```
> reco=matrix(0,nrow=19,ncol=5)
> reco=as.data.frame(reco)
> rownames(reco)=r$user
> view(reco)
>
```

The Environment pane on the right shows the 'reco' object with 19 observations and 5 variables. The 'Values' section shows the structure of the matrix.

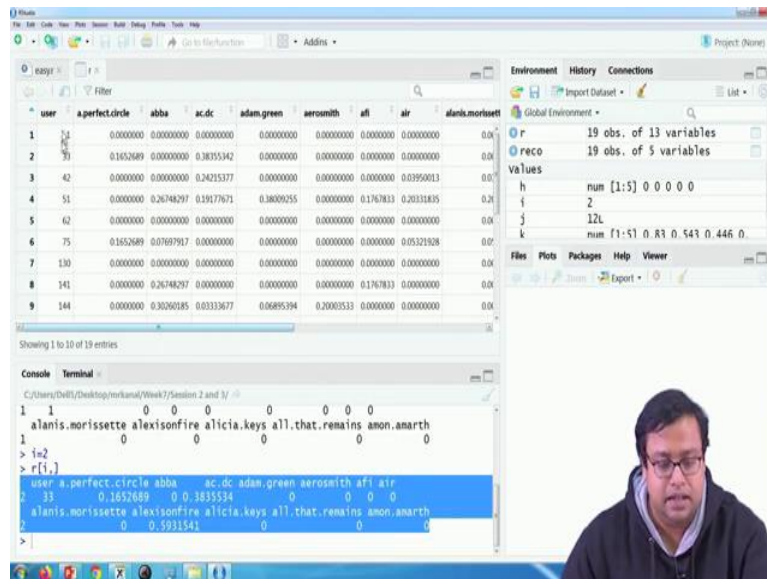
The screenshot shows the RStudio interface with R code for calculating a recommendation matrix. The code is as follows:

```
21 k=sort(a[,j],decreasing=TRUE)[2:6]
22 l=order(-a[,j])[2:6]
23 h=as.numeric(data.germany[1,(1-l)])
24
25 r[i,(j-1)]=sum(h*k)/sum(k)
26
27
28 r$user=data.germany$user
29
30 reco=matrix(0,nrow=19,ncol=5)
31 reco=as.data.frame(reco)
32 rownames(reco)=r$user
33 colnames(reco)=c("Reco1","Reco2","Reco3","Reco4","Reco5")
34
35
36 for(i in 1:19)
37   reco[i,]=colnames(data.germany)[(order(-r[i,c(2:32)])+1)][1:5]
37:55 (Dip Level) :
```

The console shows the execution of the code:

```
> reco=matrix(0,nrow=19,ncol=5)
> reco=as.data.frame(reco)
> rownames(reco)=r$user
> view(reco)
> colnames(reco)=c("Reco1","Reco2","Reco3","Reco4","Reco5")
> view(reco)
> i=1
>
```

The Environment pane on the right shows the 'reco' object with 19 observations and 5 variables. The 'Values' section shows the structure of the matrix.



These are the user names and V1, V2, V3, V4, V5 will be the name of the, so the names are Reco1, Reco2, Reco3, Reco4, Reco5 the 5 recommendations for these users and what will be the recommendations for $i = 1 : 19$, carefully see for $i = 1 : 19$ what do I do? I first I find out the corresponding entry of r. Carefully you see, let us say I am trying to find out the first guy's recommendation. So this is what I have created before the recommendation matrix. So if I just find out what is the first row, this is the user, forget about it and then these are all the values.

Now this is all 0, so this will not help me. These are all values are 0 but let us say the second guy let us say $i = 2$ so then what is r i? The user name is 33 and he has 0.16 value here 0.38 value here and 0.59 value here. So his chances of seeing alex is on fire is the highest then comes ac.DC and then comes a perfect circle. So these should be the recommendations for me, fair enough. So how will I do that? I will order this basically.

(Refer Slide Time: 31:43)

```
21 k=sort(a[,j],decreasing=TRUE)[2:6]
22 l=order(-a[,j])[2:6]
23 h=as.numeric(data.germany[l,(1+1)])
24
25 r[i,(j+1)]=sum(h/k)/sum(k)
26
27
28 r[user]=data.germany[user]
29
30 reco=matrix(0,nrow=19,ncol=5)
31 reco=as.data.frame(reco)
32 rownames(reco)=r[user]
33 colnames(reco)=c("Reco1","Reco2","Reco3","Reco4","Reco5")
34
35
36 for(i in 1:19)
37   reco[i,]=colnames(data.germany)[(order(-r[i,c(2:12))]+1)][1:5]
```

```
> order(-r[i,c(2:12)])
[1] 9 3 1 2 4 5 6 7 8 10 11
> order(-r[i,c(2:12)])+1
[1] 10 4 2 3 5 6 7 8 9 11 12
> colnames(data.germany)[(order(-r[i,c(2:12))]+1)]
[1] "alexisonfire" "ac.dc" "a.perfect.circle"
[4] "abba" "adam.green" "aerosmith"
[7] "aFi" "aIr" "alanis.morissette"
[10] "alicia.keys" "all.that.remains"
>
```

```
Reco1 Reco2 Reco3 Reco4 Reco5
1 a.perfect.circle abba ac.dc adam.green aerosmith
11 alexisonfire ac.dc a.perfect.circle abba adam.green
42 ac.dc aIr alanis.morissette a.perfect.circle abba
51 adam.green alexisonfire abba aIr alanis.morissette
62 all.that.remains a.perfect.circle abba ac.dc adam.green
75 alicia.keys a.perfect.circle abba aIr alanis.morissette
130 a.perfect.circle abba ac.dc adam.green aerosmith
141 abba aFi a.perfect.circle ac.dc adam.green
144 all.that.remains alicia.keys abba aerosmith alexisonfire
150 all.that.remains a.perfect.circle abba ac.dc adam.green
```

```
> colnames(data.germany)[(order(-r[i,c(2:12))]+1)][1:5]
[1] "alexisonfire" "ac.dc" "a.perfect.circle" "abba"
[5] "adam.green"
> for(i in 1:19)
+ reco[i,]=colnames(data.germany)[(order(-r[i,c(2:12))]+1)][1:5]
> view(reco)
>
```

So what I am doing here is order i-th cell and 2 to 12 because first entry is user, so that I am ordering in a reverse that means in a decreasing order. So this is my when $i = 2$, this is how it looks like. 9, this alexisonfire was 9. Ninth entry then third entry then first entry and so on. Then this + 1, why I am doing + 1? Because I am trying to find out the names of these movies from the original dataset.

The original dataset's first column was user, that is why + 1. So tenth cell, eleventh fourth, tenth column, fourth column, second column, third column and fifth column will give me those columns names will give me the corresponding movie names. So then what do I do?

I am saying that if the column names of this that means these are the column names, alexonfire, ac.dc, a perfect circle and then abba ,adam.green, these has been taken, these are

all 0s, so this has been taken and all these. So these three matters basically and then I am taking the top 5 only, 1 to 5 because I am giving five suggestions only. So the last two suggestions are random, the first three suggestions in this case because there were some positive value, I am getting some suggestions.

So these are my suggestions. So now if I run it for all of these guys, that populates suggestions for each of these people. So for 33, this will be the suggestions, for 42 this will be the suggestions, for 62 this will be the suggestions and so on. So that is how I create a item to item collaborative filtering. So once more, what did I do, what are the steps?

(Refer Slide Time: 33:40)

The diagram shows the process of calculating item-item similarity. It starts with a matrix of m users and n items. A specific user u is highlighted, and their ratings for items 6, 9, 11, 13, and 17 are shown as $\frac{6}{5}, \frac{9}{0}, \frac{11}{1}, \frac{13}{1}, \frac{17}{0}$. The similarity calculation for item 3 is shown as $S_3 = S_{11} + S_{13} = 0.73 + 0.42 + 0.33 = 1.48$. Another calculation shows $0.62 + 0.42 + 0.32 + 0.29 > 1.48$.

The diagram illustrates the flow from a user-item matrix to a similarity matrix and then to recommendations. It shows a matrix with m users and n items, where a user u has a rating for item i . This leads to an $n \times n$ similarity matrix. The final step shows a list of recommendations $R_u = \{i_1, i_2, \dots, i_k\}$ for user u .

First I had this, I will just do once more. First, I had m items, sorry, m users and n items, purchase data 0 1 purchase data. From there I calculated n / n similarity matrix. Then I

calculated a this is purchase data, this is recommendation matrix of m users, n items. Here the values are the recommendation scores. How the strength of the recommendation and based on that for each m users, we suggested R1 to R5 top 5 recommendations based on the scores, whichever score is higher will be the first recommendation, whichever score is lower will be the second recommendation and so on.

So these are steps that I have created in item to item collaborative filtering. So we have done it for a smaller dataset. There is a bigger dataset called Istm matrix Germany. This is available in publicly available dataset. You can use it and you can try out and find out that whether you can create a recommendation engine with the all 1,257 of these things and 286 users and 1,257 items you can create.

It will take some time. Same thing, it will take some time but you will get a result for that. So thank you very much. I will come back with user to user or user based collaborative filtering. Thank you.