

Marketing Analytics

Professor Swagato Chatterjee

Department of Ocean Engineering

Indian Institute of Technology, Kharagpur

Lecture No 4

Introduction to R Programming (Contd.)

Hello everybody welcome to the session 4 of week one in Marketing Analytics course. This is Dr. Swagato Chatterjee from VGSOM IIT Kharagpur who will be taking this course for you. So, till the last video we have actually created data set and we have saved and again read a data set from the computer. In this particular session, we will actually play with the data set little bit.

(Refer Slide Time: 00:54)

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

```
1 #Starting of Session 4
2 #Create a dataframe from vectors
3 fy <- rep(c(1999,2000,2001),3)
4 company <- c(rep("png",3),rep("hu1",3),rep("marico",3))
5 revenue <- c(11234,14567,15698,13456,14321,15643,9876,11546,13456)
6 margin <- c(11,13,12,12,12,13,11,9,14)
7 Data <- data.frame(fy, company, revenue, margin)
8
9 library(dplyr)
10
11 myresults <- Data %>% group_by(company) %>% mutate(highestMargin
12
13 highestProfitMargins <- Data %>% group_by(company) %>% summarise
14
15
16
```

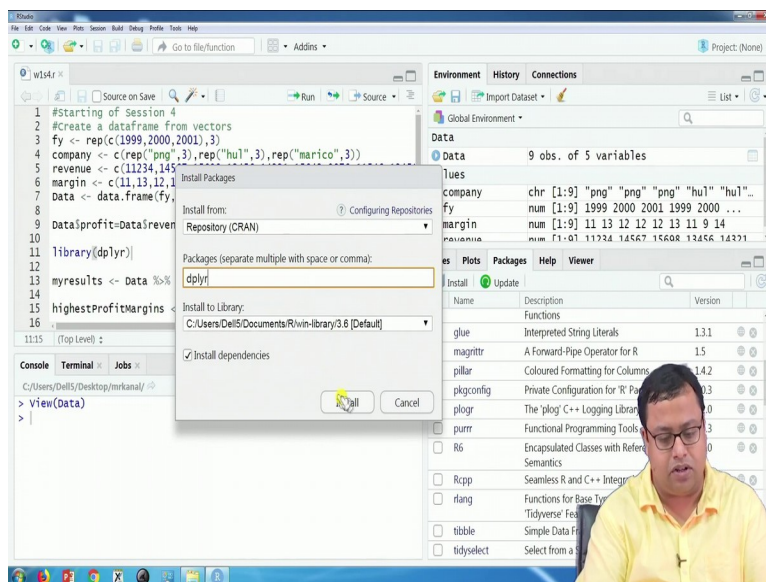
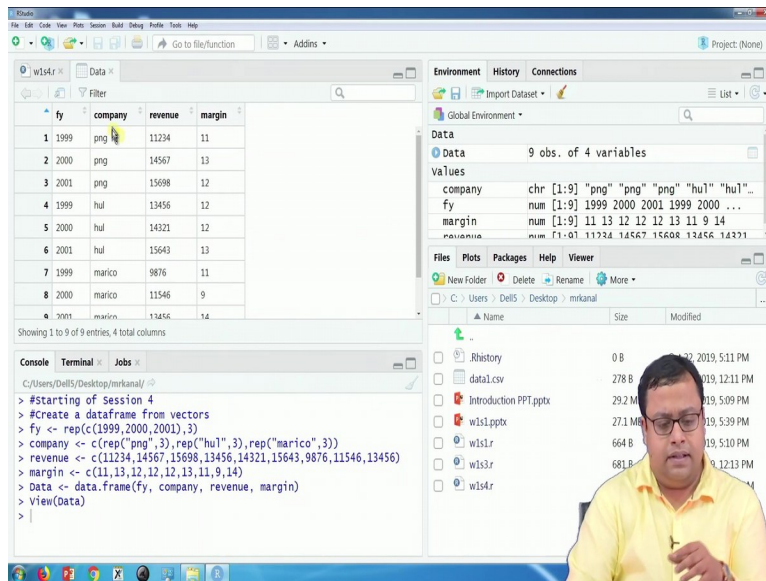
The Environment pane on the right shows the following data:

Variable	Class	Values
company	chr	[1:9] "png" "png" "png" "hu1" "hu1"...
fy	num	[1:9] 1999 2000 2001 1999 2000 ...
margin	num	[1:9] 11 13 12 12 12 13 11 9 14
revenue	num	[1:9] 11234 14567 15698 13456 14321...

The Console window at the bottom shows the execution of the code:

```
> #Starting of Session 4
> #Create a dataframe from vectors
> fy <- rep(c(1999,2000,2001),3)
> company <- c(rep("png",3),rep("hu1",3),rep("marico",3))
> revenue <- c(11234,14567,15698,13456,14321,15643,9876,11546,13456)
> margin <- c(11,13,12,12,12,13,11,9,14)
>
```

A small inset video of Professor Swagato Chatterjee is visible in the bottom right corner of the RStudio window.



So, if you open session, week one s 4, w1s4.r, so week one session 4.r file it looks like this. Again, as usual, we will actually clean our console, clean our environment and so you can click on the brush sign to clean the environment here by pressing Ctrl+L to clean the console. And then you can also remove all other files that is open in our, only w1s4.r should be open; if you are at that stage then we can start.

So, I will ask you to check lines number 3 to 6. So, quickly if you just run all these 6 lines. I will not tell you what is happening because we have already discussed about them, so can you just spend one minute on your time. Pause this particular video, spend one minute of

your time, and see that what has happened? So what has happened is that I have created 4 variables, and the company variable is a character variable, if you see line number 4. I have created company variable that has 3 “png”, then 3 “hul” and then 3 “marico” and I have also created fy which is financial year, which is 1999, 2000 and 2001 repeated 3 times. And then I have certain revenue values, certain margin values; each of them has a size of 9.

So, when I put all of this in a data frame in line number 7 that dataframe looks like this. The 1999, 2000 and 2001 repeats 3 times. For each company, there are 3 financial year, corresponding revenue, and margins are given, something like that I created. Now I will play with this data set a little bit. So there are lots of ways you can do many things. For example, the first thing that I want to do is probably calculate the actual profit. I have been given the revenue, and I have been given the margin, and I want to calculate the actual profit. So, to do that, what I will do is, so Data dollar and let us say profit. So Data dollar profit is not there right now, and I am introducing Data dollar profit as a new variable in this particular data set, and I am saying Data dollar profit is equal to, what it is, how it can be created, it is actually $(\text{revenue} * \text{margin}) / 100$, that is all.

So $(\text{data } \$ \text{ revenue}) * (\text{data } \$ \text{ margin}) / 100$. So if I just run this line, you will see in this particular thing there is a profit column that comes up. So data profit this column comes up so I can actually create new columns like this and later point of time I will show you various other things that I can do. Now, one very common thing that we try to do is to find out, let us say, in this data set how can I find out, for every, let us say, every company, what is the highest margin and what is the lowest margin out of these three things. So oftentimes, we do many things, we do like a company-wise average of sales, company wise growth rate of sales and this and that. So we break that data set in each company wise or sometimes each financial year wise and try to create something like that. So for that, there is a library called dplyr, now what is a library?

A library is something that has been written by various users, which has been validated by other users and then stored in something called repository of C, called Cran, where most of the libraries, most of the common libraries are saved. And we actually call those libraries, so library is actually nothing but a set of functions kept together. Now those functions should be of related topics like, for example, there is some library that is related to all forms of time series analysis. So, all the functions which are related to time series analysis that has been written by the writer of the particular library are stored together. Similarly, here there is a library called dplyr.

Now, oftentimes you will not know the library names, so you might also again have to go and search in Google that what you want to do, and based on that, you will find out the library and codes. So, these codes are not written by me. I have taken from someone, changed it as per my need, and kept it here. So that is what you also have to do. So, here in my next job, what I will do is, I will actually find out, for each company, what is the highest margin and what is the lowest margin. So, if this my data set. So let us say the company png, the highest margin of png is 13 the lowest margin is 11, for hul the highest margin is 13 the lowest is 12, for marico the highest is 14 the lowest is 9. So, company wise I will find it out and save in a data set. You can also do something else, and I will tell you what to do.

So, to do that the first thing that I will do is this library dplyr, I have to call that. Now, if it is not, here is a packages tab here and it actually lists out all the libraries that are already there on your computer. Now, if by chance dplyr is not there or if by running this, you get an error sign or error message, then it will mean that the dplyr package has not been installed, it is not installed in your computer. So, then you will have to install, you will have to have a sttps enabled computer secure internet connection. So, for that what you have to do is, you can click on this install button and a pop-up will come and here you have to write dplyr, so that

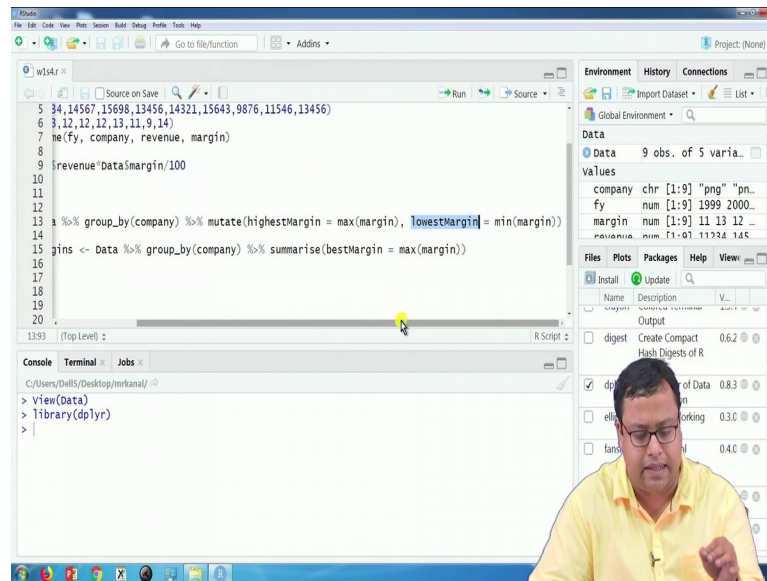
you can choose dplyr, the repository is CRAN as I told and you have to click install, and the installation is by default in the same folder where your R has been installed.

You can also write install packages, and within that with the quote “dplyr” and sometimes if you want to install multiple this thing, multiple libraries together, you can actually write dplyr comma, something else comma something else and so on. So, here we will try to install dplyr if you press an enter right now, I will not press because, for me it is already installed. But if you want to press an enter right now, you will see here a red button will come up and then you will see a lots of downloads are happening continuously 3-4 downloads or more, 15-20 downloads will be happening, and there will be small files 1 mb, 2 mb, depending on the speed of your internet, it will be pretty faster, and I hope you are seeing this particular video in 4G connection kind of stuff, and then it will be pretty fast, and lots of files will get installed.

When we install dplyr, generally, one library is dependent on another library, so when people write their codes, they actually say that this library will only run when other certain dependencies are also installed. So, all those dependencies will also get installed and downloaded, if, by chance, they are already not there on your computer. So, all of them get downloaded, and if you do not get an error message, you are good in place. So, if you by chance get an error message, there are lots of different kinds of error that can happen, and depending on what kind of error message is coming, you can post the error message in the forum, and we will try to solve that. So, once that is installed, you have to call the library, I will just run this one, and I have called the library, library dplyr has been called.

That means all the functions which are in dplyr can be used right now; they are in a usable form.

(Refer Slide Time: 9:20)



Now, this is the syntax, which is a little bit difficult, and I would want you to focus on the syntax carefully. So, line number 13; it is saying that my results are equal to. So something I do and then saves it in my results. So, my results are equal to, data that means this data, so data and then a sign which is basically percentage greater than percentage, that means it is a piping sign. It says the data and then group by company. So within data there is a company, I would say, column, so for, which is the categorical variable. So, whatever you write here has to be categorical. So group by company and then you mutate; there is 2 options, one is mutate, and another is summarize, and I will tell you what these things are.

Mutate, the highest margin is equal to the max of margin; whatever is the margin, find out the maximum of that put it in the highest margin. And whatever is margin, find out the minimum of that and put it in the lowest margin.

(Refer Slide Time: 10:37)

fy	Comp	margin	highest margin
A	A	12	13
A	A	13	13
A	A	9	13
B	B	13	17
B	B	15	17
B	B	17	17

So, what it does is something like this, so it will let us say that this is my data set, which is by company, and then there is margin, and some other things are also there. And company A, A, A, is there and company B, B, B is there, and it is coming 12, 13 and 9, let us say and 13, 15, 17, let us say. So the first thing this guy does here is that group by company, which means it will group the data set based on the company. So this is one part; this is one part; it is grouping, oftentimes your data set will not have all the A's sequentially, and all the B's sequentially, it can be A, B, A, B, A, B and so on. So it can group it all the A's together and all the B's together, irrespective of how A's and B's are written; whether A's are written in one group or A and B are haphazard, it does not matter.

Now, it will find out that what is the maximum of this margin, the maximum of this margin is 13 and the maximum of this thing is 17. So when I write that mutate, it will change the original data set, so this was the original data set. It will write, okay highest margin, I think if I am not wrong, this was highest margin is equal to the maximum of margin. So, it will write the highest margin, and the value corresponding to that will be the maximum of margin. So the maximum margin is 13, so for all of them, it will write 13; the maximum of margin is 17

here, so all of them will write 17. Similarly, for the lowest margin, it will write 9, 9, 9, and in this case, 13, 13, 13. So mutate actually change the original data set. Now, if I write instead of mutate, if I write something else, so let us say if I write summarize.

(Refer Slide Time: 12:47)

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

```
5 revenue <- c(11234,14567,15698,13456,14321,15643,9876,11546,13456)
6 margin <- c(11,13,12,12,12,13,11,9,14)
7 data <- data.frame(fy, company, revenue, margin)
8
9 Data$profit=Data$revenue*Data$margin/100
10
11 library(dplyr)
12 myresults <- Data %>% group_by(company) %>% mutate(highestMargin = max(margin), lowestMargin = min(margin))
13
14 highestProfitMargins <- Data %>% group_by(company) %>% summarise(bestMargin = max(margin))
15
16
17 #ifelse
18
19
20
```

The console window shows the following commands and output:

```
> view(Data)
> library(dplyr)
>
```

The Environment pane on the right shows the following data:

Variable	Class	Length	Summary
company	chr	[1:9]	"png" "pn...
fy	num	[1:9]	1999 2000...
margin	num	[1:9]	11 13 12 ...
revenue	num	[1:9]	11234 14567...

The screenshot shows a whiteboard with handwritten notes and a table. The notes are:

fy Comp margin

A	9
A	13
A	15
B	7
B	6
B	12

Comp
A 15
B 12

best margin

In the next line, what I have written is the highest profit margin is equal to data, group it by company, and write summarize. So summarize, what it will do, it will in this particular case, let us say, let us say I again fy and then company and then the margin and it was A, A, A and then B, B, B, and some values let us say, 9, 13, 15 and 7, 6, 12 something like that the values were there. The moment I summarize, what it will do is, it will group it first as usual, and when I summarize, it will create a new data set, which will be a summary version of the data set. Where, there will be company, and company will be A and B and that is all. And there

will be a margin; I think best margin or something like that, if I am not wrong, just I will check.

The variable is the best margin, so it will create another variable called best margin, and that value will be; in this case, it will be 15, and in the second case, it will be 12. So, it is a new data set that gets created. So all I am trying to say here is you can try out the mutate and these two things, so one is mutate, and one is summarise, and you will know what this is.

(Refer Slide Time: 14:12)

The screenshot shows the RStudio interface. The top-left pane displays a data table with the following content:

fy	company	revenue	margin	profit	
1	1999	png	11234	11	1235.74
2	2000	png	14567	13	1893.71
3	2001	png	15698	12	1883.76
4	1999	hul	13456	12	1614.72
5	2000	hul	14321	12	1718.52
6	2001	hul	15643	13	2033.59
7	1999	marico	9876	11	1086.36
8	2000	marico	11546	9	1039.14
9	2001	marico	13456	14	1883.84

The bottom-left pane shows the following R code:

```

> view(Data)
> library(dplyr)
> myresults <- data %>% group_by(company) %>% mutate(highestMargin = max(margin),
  lowestMargin = min(margin))
> view(myresults)
> highestProfitMargins <- data %>% group_by(company) %>% summarise(bestMargin = m
  ax(margin))
> view(highestProfitMargins)
> view(Data)
  
```

The right-hand pane shows the Environment tab with variables: Data (9 obs. of 5 variables), highestProf... (3 obs. of 2 variables), and myresults (9 obs. of 7 variables). The Values pane shows the company column with values: "png", "png", "png", "hu...".

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

```

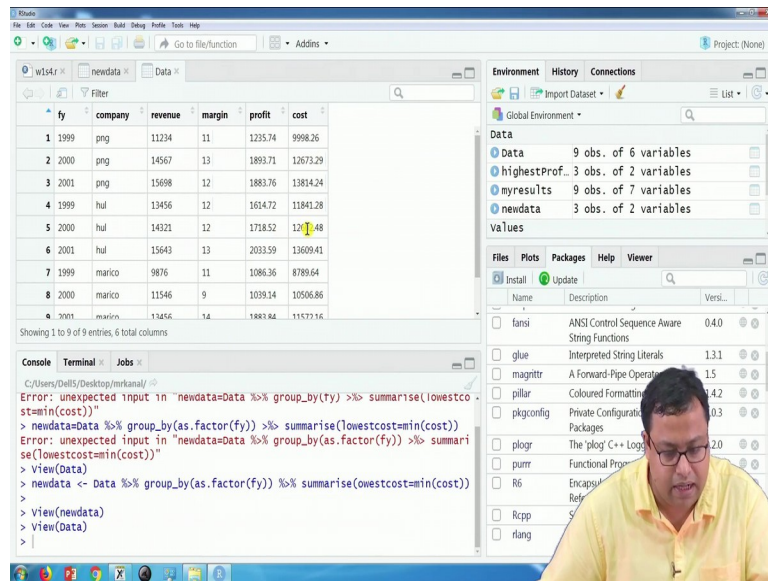
7 Data <- data.frame(fy, company, revenue, margin)
8
9 Data$profit=Data$revenue*Data$margin/100
10
11 library(dplyr)
12
13 myresults <- data %>% group_by(company) %>% mutate(highestMargin = max(marg
14
15 highestProfitMargins <- data %>% group_by(company) %>% summarise(bestMargin
16
17
18 #ifelse
19
20 #loop
21
22
  
```

The console shows the following output and errors:

```

C:/Users/Dell/Desktop/mkkanal/
> Data$cost=Data$revenue-Data$profit
> newdata=Data %>% group_by(fy) %>% summarise(lowestcost=min(cost))
Error: unexpected input in "newdata=Data %>% group_by(fy) %>% summarise(lowestco
st=min(cost))"
> newdata=Data %>% group_by(as.factor(fy)) %>% summarise(lowestcost=min(cost))
Error: unexpected input in "newdata=Data %>% group_by(as.factor(fy)) %>% summari
se(lowestcost=min(cost))"
> view(Data)
> newdata <- Data %>% group_by(as.factor(fy)) %>% summarise(bestMargin = m
  ax(margin))
  
```

The Environment tab shows variables: Data (9 obs. of 6 variables), highestProf... (3 obs. of 2 variables), and myresults (9 obs. of 7 variables). The Values pane shows the company column with values: "png", "png", "png", "hu...".



So, I will run this line, my results get created, and my results you will see highest margin is coming up, which is 13, 13, 13 for png because png's highest is 13, Hul's highest is also 13, so another 3 13, and then this guy's highest is 14, so another 3 14. And then this is for lowest margin, and if I do summarize, I get this particular thing, which is a summary version of the data, so somehow we can use dplyr for summarising this data set.

I will ask you, or I will actually run you through. If let us say, I want to find out from the same data set, what is the let us say what the highest revenue minus profit is, so highest cost. So what, in which particular this thing, in which particular, what is the highest cost of png out of the 3 or let us say, irrespective of png, forget about png. For fy, for various fy, what is the highest cost of fy 1999, 2000 and 2001. So what I do is first things first, I will just create a, in my data set, Data dollar cost, which is a new variable I am creating, is [(Data \$ revenue)-(Data dollar profit)= cost] , fair enough? And then let us say, my new data, which is a trial version is data and then this sign then I have to group by fy, I have to group by fy because for each financial year I want to know what is the highest this thing and then and then I want to summarise, summarise means I will create a new variable, summarise highest, this thing is, let us say, highest cost or lowest cost, I want to know the lowest cost.

So lowest cost is equal to minimum of cost, minimum of cost and c capital or small, this is case sensitive. So if I run this, there is an error that comes, so group by, I think group by fy, that has to be a factor. New data is equal to data this summarise minimum of cost; cost is there , c is small so, I will just once more, I will just probably copy from here and paste it here, so new data, let us say , new data is equal to data group by as factor fy and let us say lowest cost is minimum of cost, so it worked So new data is equal to this, so it has to be lowest cost, I typed it wrong, lowest cost I typed.

So whatever be the case, now here, what I am getting is as factor 1, 9, 9, 9, 9, 2000 and 2001, so these are the 3 guys, so why it happens? See I told that not always a's and b's will be in order so here 1999, another 1999 is at this place and another 1999 is in this place, so these are the 3 1999, row number one, row number 4, row number 7, they joined it together, grouped it together, and the cost is lowest for here it is 9000 , around 10000, here it is 12000, close to 12000 and here it is 8 7 8 9, and the lowest is 8 7 8 9, so it has given 11 8 7 8 9. Similarly for 2000 here it is row number 2, and then row number 5 and then row number 8 and the corresponding value is 12000, again 12000 and then 10000, so the corresponding value is 10000, and that is how you can summarise a data set.

(Refer Slide Time: 19:05)

if (check) / The want True
else / The want false.

fy	company	revenue	margin	profit	cost	
1	1999	png	11234	11	1235.74	9998.26
2	2000	png	14567	13	1893.71	12673.29
3	2001	png	15698	12	1883.76	13814.24
4	1999	hul	13456	12	1614.72	11841.28
5	2000	hul	14321	12	1718.52	12602.48
6	2001	hul	15643	13	2033.59	13609.41
7	1999	marico	9876	11	1086.36	8789.64
8	2000	marico	11546	9	1039.14	10506.86
9	2001	marico	13456	14	1883.84	11573.14

```

15 highestProfitMargins <- Data %>% group_by(company) %>% summarise(bestMargin
16
17
18 #ifelse
19
20 Data$marginhighlow = ifelse(Data$margin>10,"high","Low")
21
22 #loop
23
24 Data1=Data[Data$company=="png",]
25
26 #function
27
28
29

```

```

> View(Data)
> Data$marginhighlow = ifelse(Data$margin>10,"high","Low")
> View(Data)
> View(Data)
> Data$company=="png"
[1] TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE
> Data1=Data[Data$company=="png",]
> View(Data1)

```

So we will do a little bit more, we will learn something called ifelse function. So ifelse is something like what you do in excel, so if function, how do you write in if function in excel, you write it like this way, you write is equal to if and then there are 3 syntaxes basically. So one is the check, what you want to check, the logic, if the logic is true the result when true and this is the result when false, that is how the if the function is written in excel if you have tried it. Similar things we will do it here, the only thing that becomes is, it becomes ifelse.

So I will write it in this way ifelse function, so let us say I want to say in this data tab if the margin is higher than 10, I will say that high, high margin, it is high margin, so I will create a new column which is called margin high low and if the value of margin is higher than 10, I will say high, if the margin is lower than 10 I will say low, so I will write that. So I am just saying data dollar margin high-low that means it is a new variable called margin high low, which is equal to ifelse, under ifelse there are 3 syntaxes total. So I will just put 2 comas, so that it becomes easier to write. Two commas I have written. First thing before comma is the syntax, which will check whether the margin is higher than ten or not, so my syntaxes data dollar margin whether this guy is higher than 10 or not. The second syntax is what the second syntax is basically the high so this is high and the third syntax is what? When it is not greater than 10 then what result will come, so I am saying it is low.

See high and low I have written in quotes because these are text, so data dollar margin high low I am creating a new variable in my data set is equal to ifelse, ifelse is the function it has 3 syntax, first is the check, second is the value that comes when the check is right, and the third is the when the value is false. So if I just run this one , I have created another column in data set c , there are highs and some lows, so these guys coming low because this is the only guy who is where margin is 9 which is less than 10. You can also have multiple this thing, so let us say if it is greater than, smaller than ten or let us say greater than fifteen, I will say very high, but 10 to 15 I will say high and otherwise low. So you can do all these ifs and buts, and

whether it is greater than or lower than you can have multiple things, you can join them through and, and or and etc.

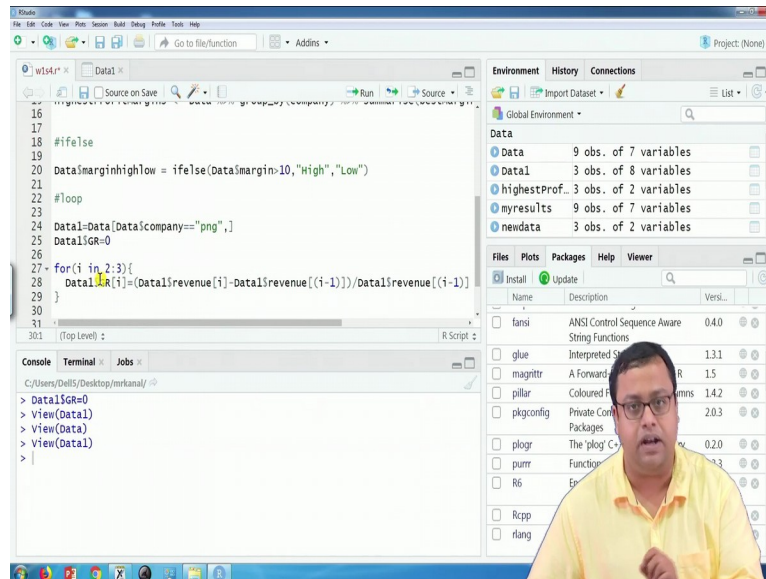
What we did in the previous videos that you can do it here as well. Now I will teach another thing which is called a loop this is an important thing. Loop means that same thing gets repeated continuously. So let us say, I am trying to do it in this way that I will start from the first row, end at the last row and from the first row, I will start from the first row, end of the last row and I will say that every this thing how much jump has happens. So how much was the revenue growth that has happened? So revenue growth for png is let us say from this to this, so I will do it fast forward only png, let us make it simple. So from this to this how much growth has happened in percentage term and from this to this how much growth has happened in percentage term.

So to do that for only png, not anybody else, the first thing I will do is, I will create a data set which is a subset of the data which has only png, so data dollar company double is equal to png. So those who have a little bit of knowledge about coding before you will know that for check of equality, if you do not want to assign something but if you want to check of equality then you have to give double equal to so instead of greater than, smaller than check whether it is higher or lower, double equal to checks whether it is absolutely same or not. Now check what I have written here, I have written that I am subsetting this data set called data, the name of the data is data so I am subsetting it. I have written nothing after the comma that means all the columns I am taking. But I have written something before the comma, what did I write? I wrote only this part, if I just copied this part and paste it here.

I am getting first 3 values as true and rest of the values as false because the first 2 guys are for png, and the next 6 guys are false because they are not of png. Now, if I put here, subset is based on this, then the first 3 rows will be taken and all other rows will be left aside. And

all the columns will be taken because nothing is written after the coma. So if I run this, I get a data set which has only png, fair enough.

(Refer Slide Time: 24:49)



```
16
17
18 #ifelse
19
20 Data$marginhighlow = ifelse(Data$margin>10,"High","Low")
21
22 #loop
23
24 Data1=Data[Data$company=="png",]
25 Data1$GR=0
26
27 for(i in 2:3){
28   Data1$GR[i]=(Data1$revenue[i]-Data1$revenue[(i-1)])/Data1$revenue[(i-1)]
29 }
30
31
32 (Top Level)
R Script
```

Environment History Connections

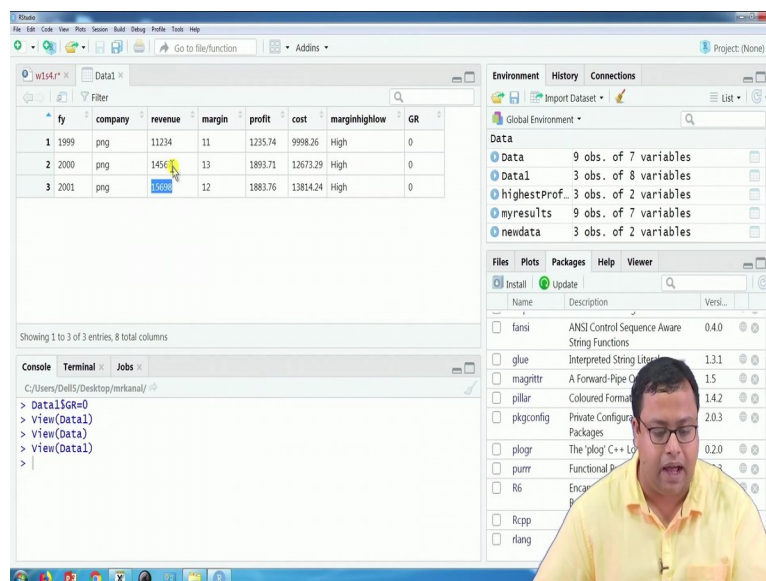
Global Environment

- Data 9 obs. of 7 variables
- Data1 3 obs. of 8 variables
- highestProf 3 obs. of 2 variables
- myresults 9 obs. of 7 variables
- newdata 3 obs. of 2 variables

Files Plots Packages Help Viewer

Console Terminal Jobs

```
> Data1$GR=0
> View(Data1)
> View(Data)
> View(Data1)
> |
```



fy	company	revenue	margin	profit	cost	marginhighlow	GR	
1	1999	png	11234	11	1235.74	9998.26	High	0
2	2000	png	1456	13	1893.71	12673.29	High	0
3	2001	png	1569	12	1883.76	13814.24	High	0

Environment History Connections

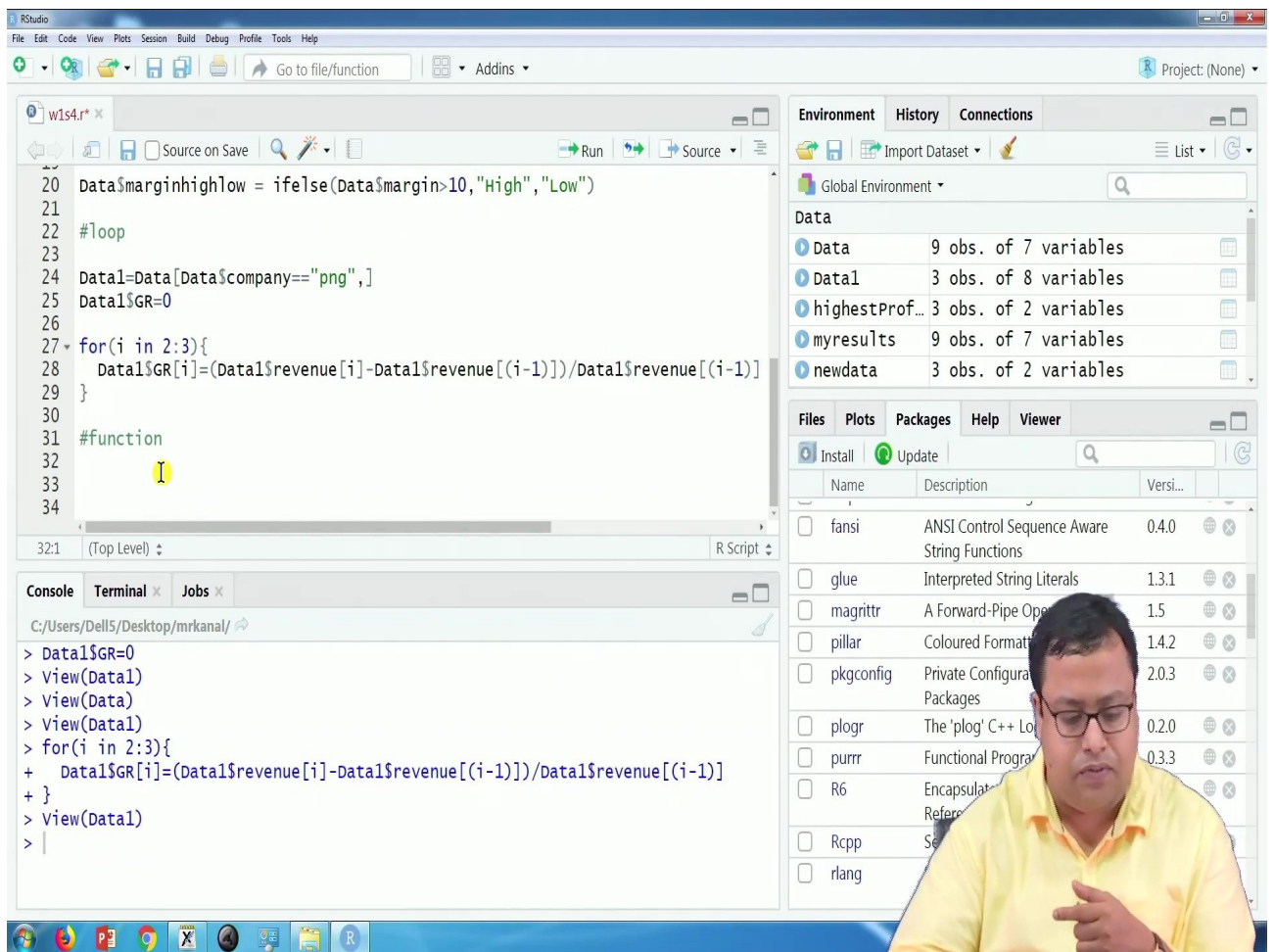
Global Environment

- Data 9 obs. of 7 variables
- Data1 3 obs. of 8 variables
- highestProf 3 obs. of 2 variables
- myresults 9 obs. of 7 variables
- newdata 3 obs. of 2 variables

Files Plots Packages Help Viewer

Console Terminal Jobs

```
> Data1$GR=0
> View(Data1)
> View(Data)
> View(Data1)
> |
```

Now I will create another column which is a growth rate, and I will say that growth rate is nothing but the previous versus now divided by the previous. So I will write it this way. So, let us say, I am writing (Data1 \$ gr=NA). So I am introducing or is equal to 0 let us say, I am introducing something which is currently 0 and then I will replace this, now I am writing a loop.

Now remember this loop will start from 2, go up to 3, so I am writing for i = 2 to 3, just carefully see what I am writing, for i =2-3, so i is an index which starts at 2 ends with 3 and then I write a bracket so you do not write it right now and when you try to practice it by writing after I have written everything you just try to replicate that. So for i(2, 3), what happens? I will first find out, so at the first when the loop starts i's value is 2, so i's value is 2 means, then data dollar, then say if I write revenue i that is what, i's value is 2, so data dollar

revenue is, sorry not data dollar revenue, data one dollar revenue, so data one dollar revenue is what, data one dollar revenue is this particular column, and here i is equal to 2 that means second entry, so this is what I have written.

```
for(i in 2:3){  
  
Data1$GR[i]= (Data1$revenue[i] - Data1$revenue[(i-1)])/ Data1$revenue[(i-1)]  
  
}
```

So, current value minus the previous value divided by the previous value. That is the growth rate, so how much is the growth rate, where will it be saved. This will get saved in this value whatever written, data one dollar gr i, that means when i is equal to 2, 2 minus one by one will get saved in 2, and when i is equal to 3 , 3 minus 2 by 2 will get saved into 3, the third entry. So for this one, 1 4 1 6 7 by 1 1 2 3 4 divided by 1 1 2 3 4 will be saved here.

And then next, 1 5 6 9 8 by 1 4 5 6 7 divided by 1 4 5 6 7 will get saved here. So this is how I am calculating a loop. Now to run this loop what do I have to do, I have to select the whole thing, you can see that there is a small drop-down button coming up at 27 so it is saying that loop starts a loop ends, if you click on that, 27 to 29 gets collapsed, if you again click on that it comes up. So all you have to do is whatever gets collapsed together, you have to run the whole thing together. So I am selecting the whole thing together, and I am running it, and it has run, and if I now want to see data one, you see that the percentage changes or probably the point values are given. So the first jump was 29 percent, 29.66 percent, the second jump was 7.7 percent. You can try to run a loop on your own at a later point of time; we will try to see what loop is.

And the last part of this particular thing is called a function.

(Refer Slide Time: 29.11)

The diagram illustrates a function and a data flow process. At the top, a box labeled 'fn' has an arrow labeled 'Inputs' pointing into it from the left and an arrow labeled 'output' pointing out of it to the right. Above the 'fn' box is a circle containing the word 'Job'. Below the 'fn' box is another circle containing the word 'jobs'. Below the 'jobs' circle is a box containing the mathematical expression $f(x) \rightarrow y$. To the left of this box is a circle containing the variable x .

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

```

20 Data$margin$highlow = ifelse(Data$margin>10,"high","Low")
21
22 #loop
23
24 Data1=Data[Data$company=="png",]
25 Data1$GR=0
26
27 for(i in 2:3){
28   Data1$GR[i]=(Data1$revenue[i]-Data1$revenue[(i-1)])/Data1$revenue[(i-1)]
29 }
30
31 #function
32 |
33
34

```

The console output shows the execution of the code:

```

> Data1$GR=0
> View(Data1)
> View(Data)
> View(Data)
> for(i in 2:3){
+   Data1$GR[i]=(Data1$revenue[i]-Data1$revenue[(i-1)])/Data1$revenue[(i-1)]
+ }
> View(Data)
> View(Data)
>

```

The diagram shows a mathematical formula for normalizing revenue. It starts with x and y (where $y = f(x)$). Below this, a list of values is shown: $\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$. The formula for normalization is:

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

So we have written, we have used lots of function which has been already prebuilt, but we have to write a function, if I write a function like this, the function is something like this, where there will be certain inputs, it can be more than one inputs, and there will be certain outputs against more than one outputs, and there will be certain jobs that are done by this particular function. So that is how a function works, there will be some input, and there will be some output, and there will be some certain jobs here. So now the job that you define is irrespective of, so you have to say so function if you remember we write it f of x is a function, so x is the input and f of x is the job that it does and gives something as an output which is y , so that is how we will also write a function. Now x can be more than one item. For example, let us say if I ask you that do one thing, write a function where it will take any column as its input, and it will do one thing easily, it will find out, let us say the z value of the column. So those who have done statistics, I hope that you have an idea, or it will give the normalized version of the column. This is very common that we do, so what is normalized thing, the normalizing thing is that if I have an old z , if I have x the normalized form of x is nothing but x is a variable let say x has lots of values from 1 2 5 7 9 11 15 whatever and I want to compress those values with 0 and 1 in linear order.

I will interpolate them between 0 and one, so what I write is $x = (x - \min(x)) / (\max(x) - \min(x))$, something like this. So this is something that we try to write and this is a function that I will write, which will take any x and give this as an output. So what will I write, I have to first say that it is a normalize I do not know is equal to any name you can give another name, does not matter, is equal to a function of x , it is $f(x)$, remember in mathematics we write it like this way, so $y=f(x)$ such that it is 5 when $x > 2$, it is e^x when $x < 2$ or something like that, so this is how we write. So first we write is $y=f(x)$

(Refer Slide Time: 31.56)

Handwritten equation on the whiteboard:

$$y = f(x, z) = f(x)$$

```

23
24 Data1=Data[Data$company=="png",]
25 Data1$GR=0
26
27 for(i in 2:3){
28   Data1$GR[i]=(Data1$revenue[i]-Data1$revenue[(i-1)])/Data1$revenue[(i-1)]
29 }
30
31 #function
32
33 normalize=function(x){
34   abcd=(x-min(x))/(max(x)-min(x))
35   return(abcd)
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

```

Environment pane:

Name	Description	Vers...
company	chr [1:9] "png" "png" "png" "hu...	
fy	num [1:9] 1999 2000 2001 1999 2...	
i	3L	
margin	num [1:9] 11 13 12 12 13 11 ...	
revenue	num [1:9] 11234 14567 15698 134...	

Files pane:

Name	Description	Vers...
fansi	ANSI Control Sequence Aware String Functions	0.4.0
glue	Interpreted String Literals	1.3.1
magrittr	A Forward-Pipe Operator	1.5
pillar	Coloured Formatting for R	1.2
pkgconfig	Private Configuration for R Packages	0.2.3
plgr	The 'plgr' C++ Logging Library	0.1.0
purrr	Functional Programming	0.3.2
R6	Encapsulated Classes with Reference Semantics	0.4.5
Rcpp	Seamless R and C++ Interoperability	0.12.1
rlang	R and C++ Interoperability	0.4.1

```

13 myresults <- Data %>% group_by(company) %>% mutate(highestMargin = max(marg
14
15 highestProfitMargins <- Data %>% group_by(company) %>% summarise(bestMargin
16
17
18 #ifelse
19
20 Data$marginhighlow = ifelse(Data$margin>10,"High","Low")
21
22 #loop
23
24 Data1=Data[Data$company=="png",]
25 Data1$GR=0
26
27 for(i in 2:3){
28   Data1$GR[i]=(Data1$revenue[i]-Data1$revenue[(i-1)])/Data1$revenue[(i-1)]
29 }
30
31 #function
32
33 normalize=function(x){
34   abcd=(x-min(x))/(max(x)-min(x))
35   return(abcd)
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

```

Environment pane:

Name	Description	Vers...
Data	9 obs. of 8 variables	
Data1	3 obs. of 8 variables	
highestProf...	3 obs. of 2 variables	
myresults	9 obs. of 7 variables	
newdata	3 obs. of 2 variables	

Files pane:

Name	Description	Vers...
fansi	ANSI Control Sequence Aware String Functions	0.4.0
glue	Interpreted String Literals	1.3.1
magrittr	A Forward-Pipe Operator	1.5
pillar	Coloured Formatting for R	1.2
pkgconfig	Private Configuration for R Packages	0.2.3
plgr	The 'plgr' C++ Logging Library	0.1.0
purrr	Functional Programming	0.3.2
R6	Encapsulated Classes with Reference Semantics	0.4.5
Rcpp	Seamless R and C++ Interoperability	0.12.1
rlang	R and C++ Interoperability	0.4.1

Console:

```

> Data$profit
[1] 1235.74 1893.71 1883.76 1614.72 1718.52 2033.59 1086.36 1039.14 1883.84
> normalize(Data$profit)
[1] 0.19769722 0.85933933 0.84933380 0.57879230 0.68317160 1.00000000
[7] 0.04748353 0.00000000 0.84941425
> Data$normprof=normalize(Data$profit)
> View(Data)

```

So similarly, if there are, if y is a function of multiple values then I have to write y is function of, so if y is function of multiple values then I have to write $y = f(x, z)$, let us say.

The moment I write 2, I know that there are 2 inputs that I have to give, if I write only $f(x)$, I know that I have to give one input, similarly here depending on how many input we want to give. We have to write x or x comma y or x comma whatever, x one comma x 2 comma x 3 etc. So here I want only one input, so normalize is equal to a function of x , and then what do I want, I want $(x - \min(x)) / (\max(x) - \min(x))$, this is what I want. And I want to save it let us say in some name, some name, `abcd`, `abcd` is equal to this and it is good practice when you write a function because often times there are situations where you have to write multiple lines of code within function, it is not straight forward one liner, you have to do these then that, and these then that, and lots of things and then you get some value and you want to return that value. So it is very important to say that what do you want to return, so I want to return within bracket `abcd`.

See here also like for loop, there is a drop-down thing has come up so the whole thing that is coming up so I will choose the whole stuff and run this, the moment I run this you will see here there is function called `normalize` get saved, now this is a function, which name is `normalized`, I have saved this function, but I have not used this function till now. Now, if I want to use it, let us say there is data `dollar profit`, is a profit, and I want to normalise it between 0 and 1. So I will write `normalise` and within bracket `data dollar profit`, so take this particular value and normalize them, and this is the output. See all the values are between 0 and one, the lowest value is 0, and the highest value is one, and other values are between 0 and one, and it is also important that I have not printed I have not saved it somewhere so if you want to save it in the data set you have to write `data dollar normprof` that is an `normalize profit` is nothing but this thing so then it gets saved in your data set. So in the dataset you get the `normprof` while it is 0 and 1. So all I have done here in this particular video is to create a

data set and then using library dplyr to create summary or mutate of the data set, and then we used ifelse, loop and function, these are very small, small basic examples but we will use a little bit of complex version when we actually do and solve certain problems. Thank you very much for being patient in terms of hearing this particular video, in terms of practicing video, we will meet in the next video.