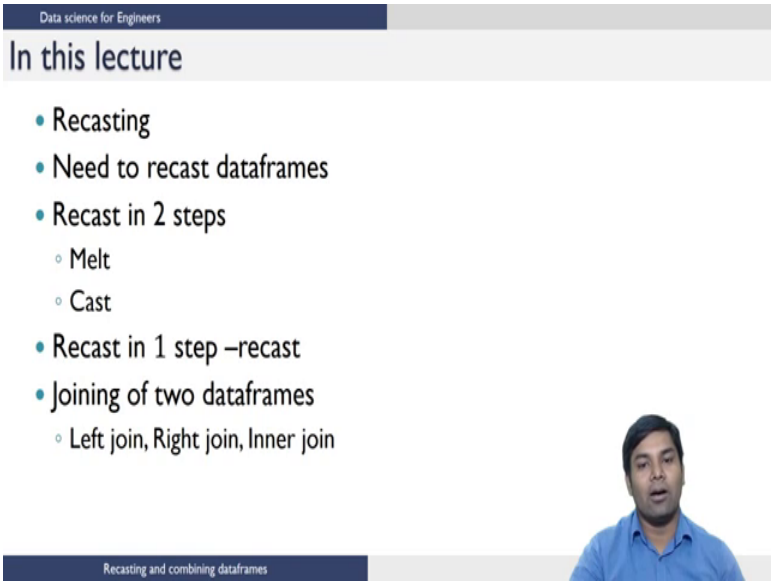
Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 06
Recasting and joining of dataframes

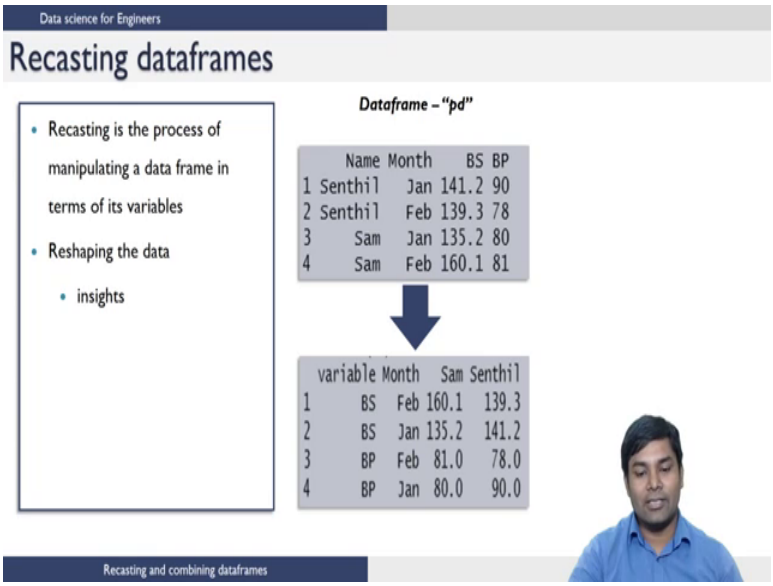
Welcome to the lecture 5, in the r module of the course data science engineers. In the previous lectures, we have seen how to create data frames? How to access rows and columns of data frames? How to add rows and columns to existing data frame? And so on. Here we will look at more sophisticated operations on data frames, such as recasting and joining of data frames.

(Refer Slide Time: 00:42)



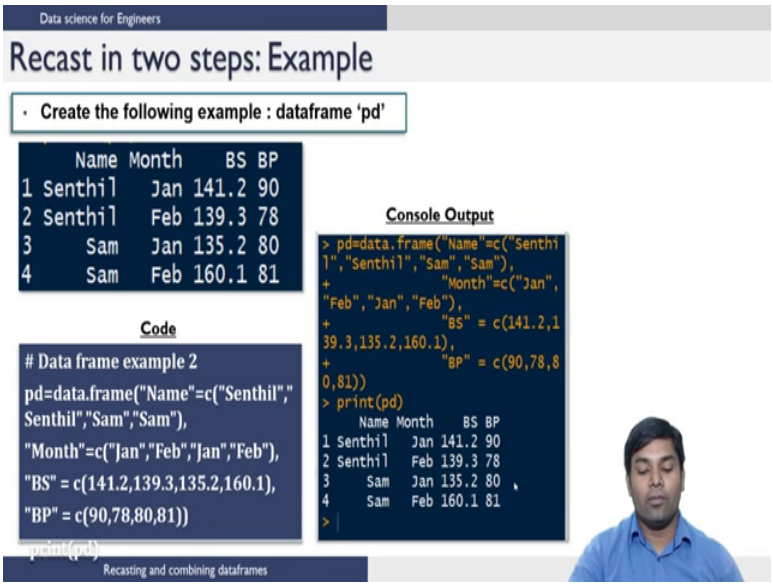
In this lecture we are going to first define, what is recasting of a data frame means? Why do one need to recast the data frames? How the recasting can be done in 2 steps using melt and cast command? How the recasting can be done in a single step using recast command? And how to join 2 data frames using left join right join and inner join functions of d player package in r?

(Refer Slide Time: 01:10)



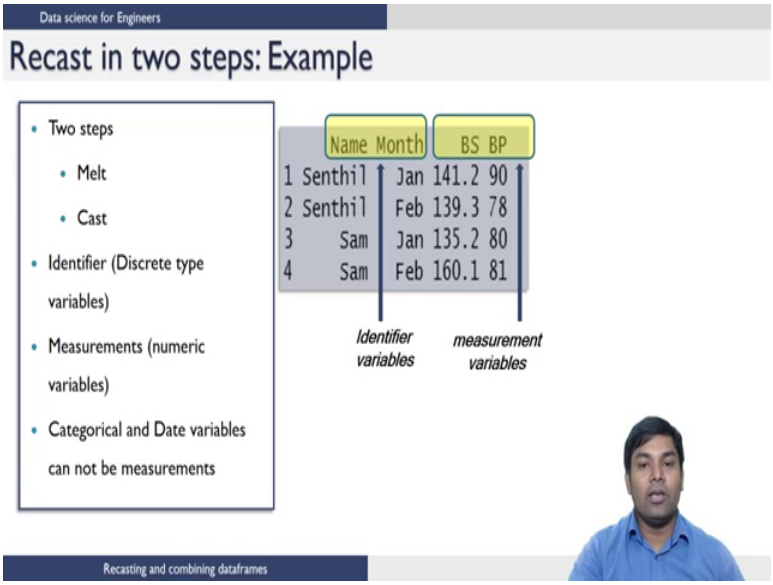
Let us first see what is recasting of data frames means, requesting as a process of manipulating data free in terms of it is variables. Why do want wants to recast the data frames? The answer is recasting helps in reshaping the data which could bring more insights on the data, when it is seen from the different perspective. Let us take a data frame which is created in the last lecture, we have the data frame name p d which has column name month, blood sugar and blood pressure. So now, you want to convert this data frame into the other form which is shown below, where you have blood sugar and blood pressure as the variables of importance to you and this involves an operation which is called recasting, this recasting is demonstrated using an example here.

(Refer Slide Time: 02:01)



So, in order to do recasting we have to have a data frame, which is the following which is shown in screen. To create this data frame, you can use the code that is displayed in screen this one and when you use this code and execute this, you will see the data frame which is shown here. Since we have the data frame now, we can see how to recast the existing data frame into the form which we want.

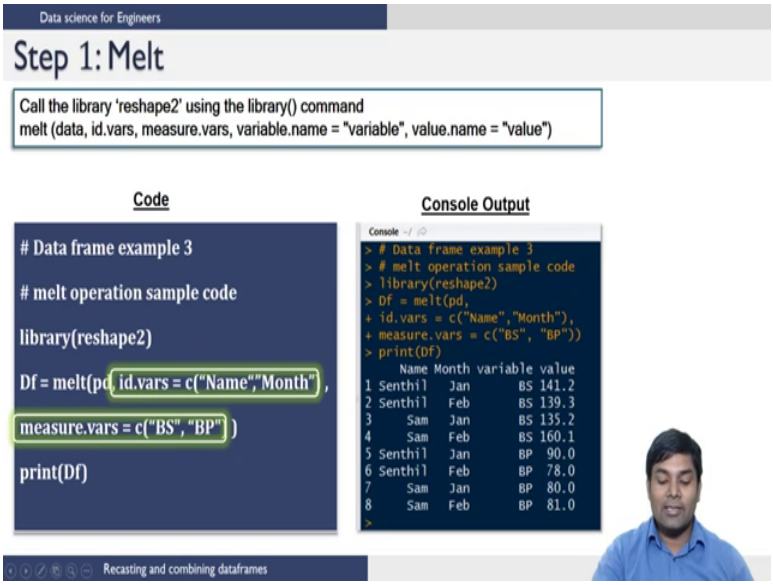
(Refer Slide Time: 02:36)



Let us see an example to demonstrate this, how to recast the data frame into another form, using 2 steps first one is melt and the second one is cast. This is the data from you

have when you want to use melt and cast command to recast, the data frame you need to identify what are called identifier variables and measurement variables of your data frame. The rules for indentifying this identifying variables are, most of the discrete type variables can be identifier variables, measure the numeric variables can be measurement variables and there are certain rules for the measurement variables such as, categorical and date variables cannot be measurement variables. So, the key idea is from the data frame, you have to identify what are called identifier variables? And what are called measurement variables? Once you have identify this identifier variables and measurement variables, you are ready to do the melt operation which you are going to see now.

(Refer Slide Time: 03:28)

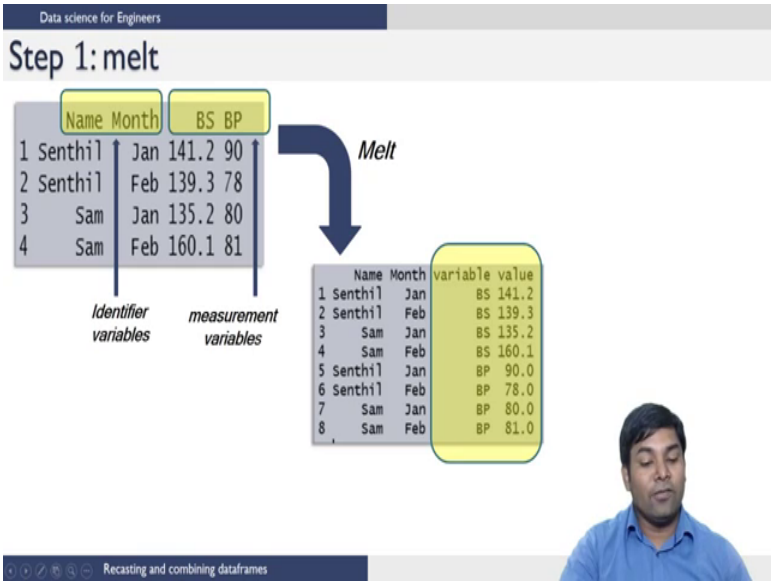


This melt command is available in the reshape 2 library, here is the first time we are loading another library to perform some operations. In the pre-course material we have given you how to install packages and this library command helps you do load the packages, that are already installed and this is the syntax of the melt comment. For the melt command you have to give the data frame as first argument and you have to specify what are the identifier variables in your data frame , and you have to also specify what are the measurement variables in your data frame and these are the default variable and value arguments that, are generated when the melt command is executed. To do this melting operation, we can use this code you first have to load this library reshape 2 which contains this functions melt and cast.

So, the syntax we have seen, melt you need to pass the data frame which you want to melt and you have to also specify, what are the identifier variables in the data frame your pass, what are the measurement variables that you are passing? Once you do this initial data frame will become like this, what it has done is, since this name and month or given as I d variables, there as it is and measurement variables BS and BP are now start under a column by name variable, as you can see here and the values of them are stored in another column, which is named as value.

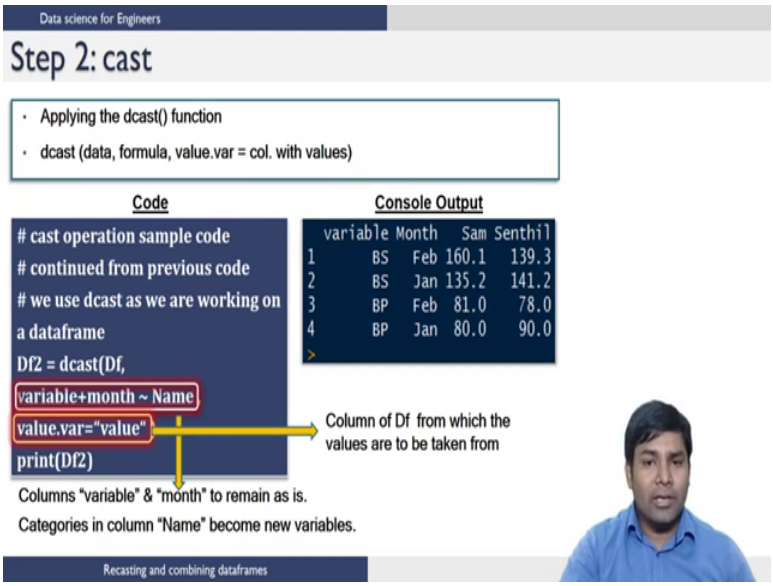
So, when the melt command is executed, it will take this I d variables and keep them assist and then convert the measure variables into, one single column which is given by variable and stores the values of those variables, in another column by name value, that is what when you say in this syntax variable dot name is variable and value dot name is value means.

(Refer Slide Time: 05:34)



So, the columns which carries this measurement variables is named as variable and which wholes the values of the measurement variable is named as value. So, this is the first step identifier variables and measurement variables of the data frame, and uses the melt command to melt the data frame to get to this structure.

(Refer Slide Time: 05:44)

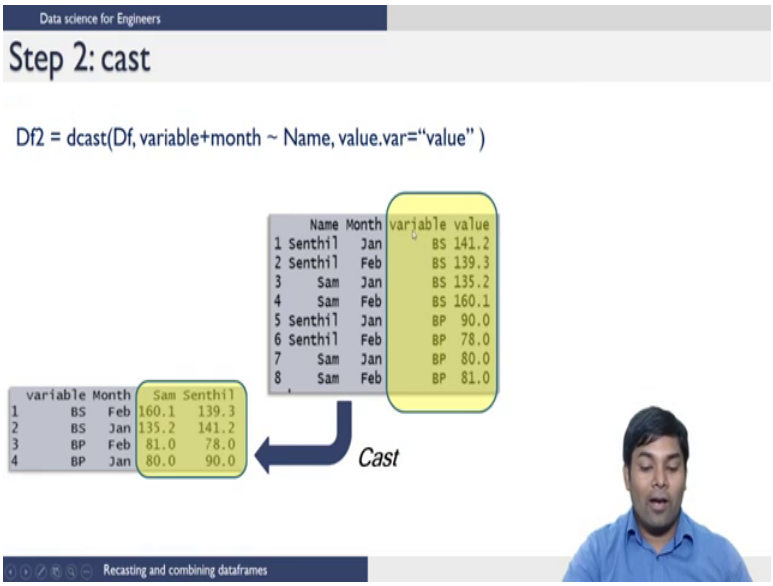


Next step is the cast, since we are using data frame here, we use the function d cast this d cast function is also available in reshape 2 library the syntax for d cast is as follows, the d cast command takes in the data frame, which you want to d cast and the formula which will explain for this case what it is? And value dot var. So, you have to specify the columns from which the values to be taken from when you are d casting.

Let us see the example our case, here you have a data frame D f which you already melted. Now, you are creating another data frame D f 2 by using d cast command, this is the data frame which you are passing that is D f and this is the formula. What does it say? I want to have this variable and month as constant, because you want blood sugar and blood pressure to be your variables of importance and then, you have to convert the name variable into 2 columns are, how many of a columns depending upon the number of categories in the name.

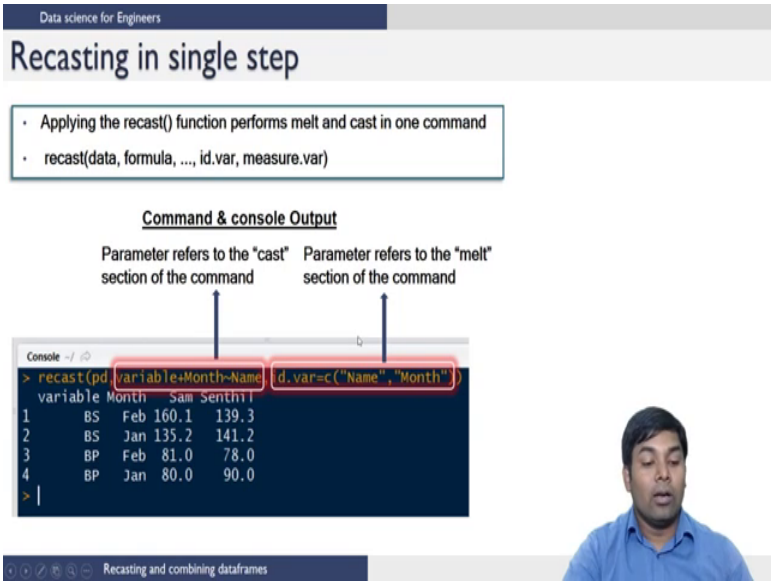
That is what this formula explains, columns variable and month remain as it is and the categories in the name becomes new variable. We have 2 categories in this example, which are Sam and Senthil and they become the new columns, that are new variables and the values for those variables has to be picked from the column value, that is what this value dot variable suggests. Once this operation is done, if you print the data frame this is how you will get the data frame in your required format.

(Refer Slide Time: 07:15)



So, you have this melted data frame from this, when you apply d cast function you pass in this data frame and you say variable and month are the ones, which you want to have it as constant, that are the left side of the formula and in the to the right of the formula you have name. So, in this name column we have 2 categories Sam and Senthil and those will be created as 2 new columns and the values for those columns, have to be taken from the value column of the melted data frame, that is how the cast command works.

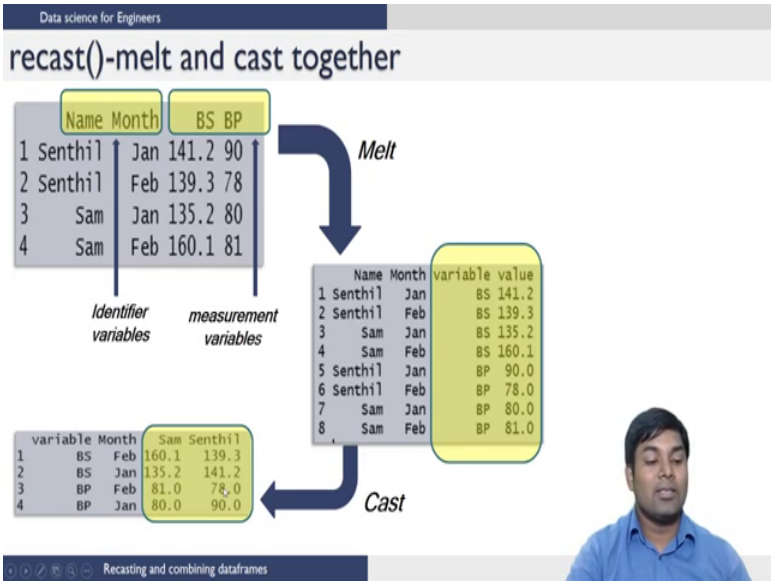
(Refer Slide Time: 07:52)



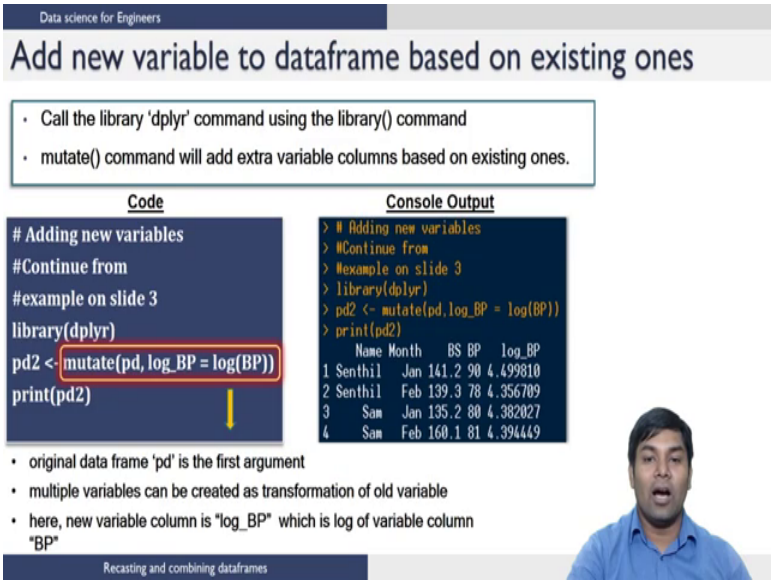
Now, let us see how to do this recasting in a single step. So, recasting can perform in a single step, using recast function the syntax for this is as follows, recast you have to give the data and the formula and we have to also give id variables and measurement variables. So, if you can see these input arguments, it takes the input arguments of both melt and cast as you can see in this command here.

So, recast command takes the data frame and it also takes the formula, this is the parameter that refers to the cast section of the command and this is the parameter, that refers to the melt section of the command. What we have seen in the melt, we have to specify what are the I d variables and the measurement variables. So, when you specify only I d variables, the rest of the variables are defaultly taken as the measurement variables. So, that is why we did not specify measurement variables here, you can also specify the measurement variables, as we can see from the syntax. Now when you execute this command, it will melt and it also cast and it will print the casted data frame as shown in this screen below. Next with this, we can see that melt and cast operations can be done together using the recast command.

(Refer Slide Time: 09:05)



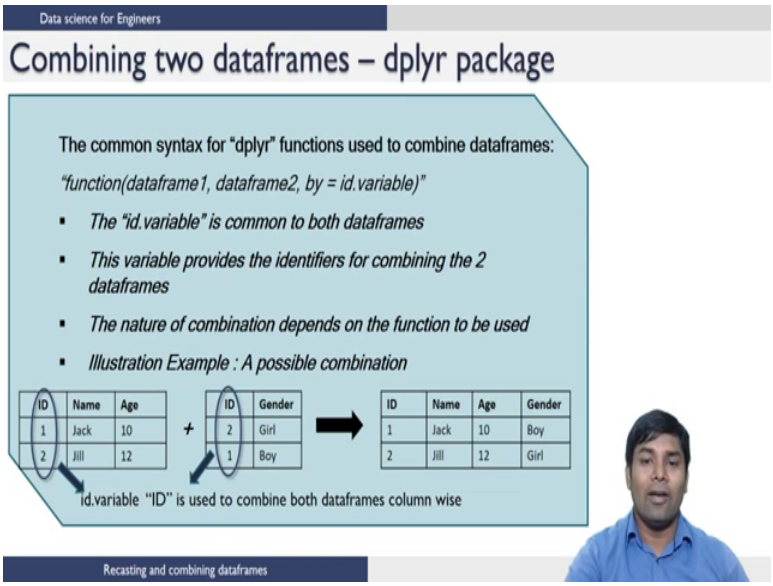
(Refer Slide Time: 09:16)



Next, we see how to create a new variable, that is a function of already existing variable, using the mutate command. Sometimes it is essential to have a translated or the function are variable, which is created from the existing variables. In this case, let us assume logarithm of BP value is something which is giving us more insight about the data. How do you create a new variable which carries the logarithm value of the blood pressure from the existing blood pressure value? Is the question..

So now, how to do that is you have to load the library dplyr, you can use mutate command and you need to pass the data frame and you have to say, you have to create new column which is carrying the values of logarithm the existing column BP. Now, if you print this p d 2 you can see that, there is another variable that is logarithm of BP that is created and you have the corresponding values of it. Now, let us look how to join 2 data frames it is very important in terms of data analysis, because you will get part of the data from one source and the part of the data from other source, when we want to match these 2 data, which are having some common I ds, how do you do this? Is the question.

(Refer Slide Time: 10:33)



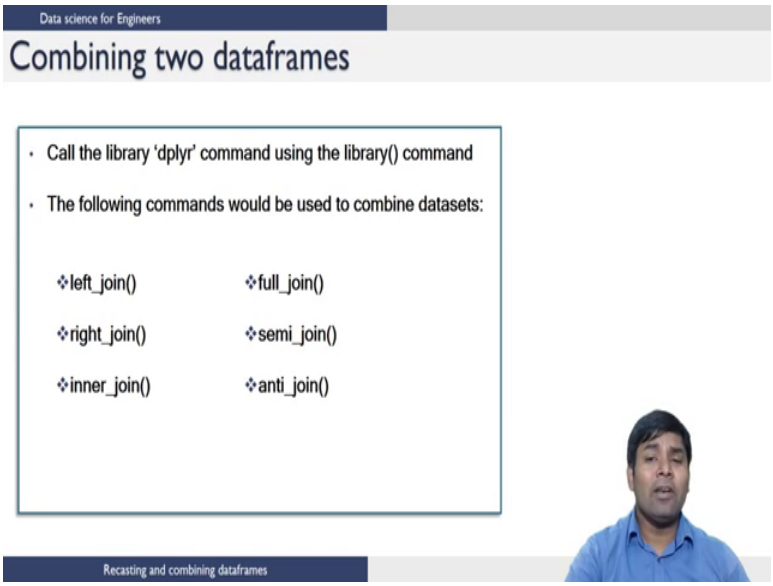
So, this combining of data frames can be done using, dplyr package the general syntax of the dplyr is as follows, you need to have a function which could be either left join, right join, inner join and so on. And you need to pass the first data frame and the second data frame, because you want to do joining of this 2 data frames and you have to specify, by which I d variable you have to join this 2 data frames.

So, here the I d variable is common to both data frames; that means, you have to have that variable in both data frames, which you want to combine and this variable provides the identifiers for combining the 2 data frames and the nature of combination depends upon the function that is being used. We will see some examples and (Refer Slide Time: 11:23) example let us see this one, we have one data frame which carries I d name and age. We have one and 2 as I ds here, name as Jack and Jill whose ages are 10 and 12 at a suppose, we have another data frame which has his I ds in the reverse order, I d 2 I d 1 and gender is girl and boy and this is output you want to get, let us say you want to merge these 2 data frames using some function either left join or right join are something.

So, that you will get the data frame which contains information in both the individual data frames for example, you can see the I ds are the common variables or the identifiers variables that are common to both data frames and we are using this I d variable, to combine this 2 data frames 1 and 2. So, we have 1 Jack and for 1 we have boy and we

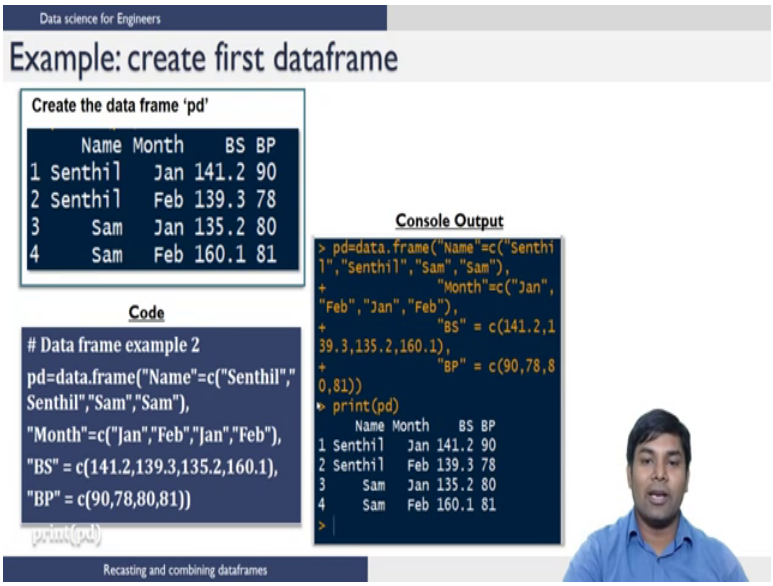
have age of Jack as 10 and that is also been taken care, and you have Jill and the I d variable of Jill is 2. So, we will have 2 Jill age and the gender this is one example, how the merging and combining the data frames happens? Now, let us look deep into the different functions, that available in the dplyr package to combine 2 data frames.

(Refer Slide Time: 12:39)



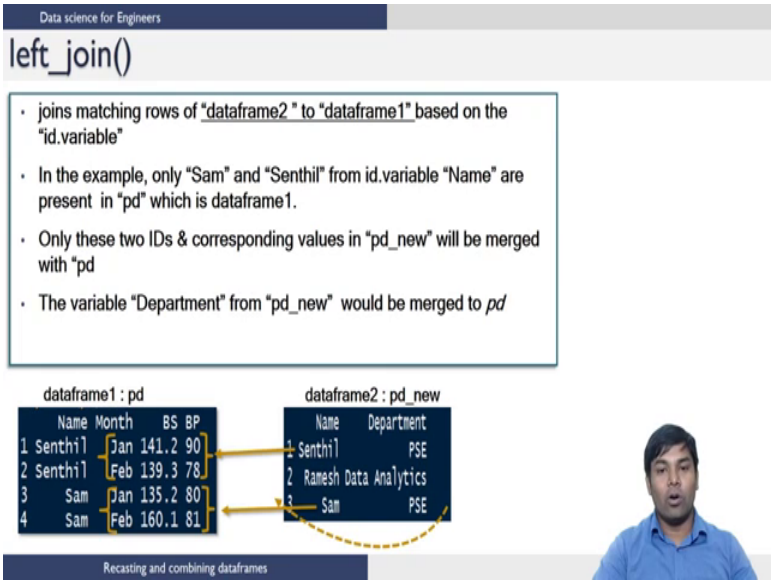
There are several functions that are available in the dplyr package to combine data frames, few of them are left join, right join and inner join and there are full join, semi join and anti-join. In this lecture, what we have going to see are the first 3 left join, right join and inner join. We will leave the audience as an exercise, to understand what full join, semi join and anti-join does in combining the data frames.

(Refer Slide Time: 13:08)



Let us illustrate joining of data frames by creating 2 data frames first, let us first create this data frame p d, this can be created using the code shown here and when you print that, you can see the output as share we have 2 names Senthil and Sam, we have 2 months Jan and February, we have blood sugar and blood pressure values of those variables. Now, we are taking another data frame which contains 3 names Senthil Ramesh and Sam and the other column carries the department, where they are working. So, Senthil and Sam is working in P S E and Ramesh is working in data analytics, to create this data frame you can use this code and when you print this data frame, you can see the result in the console has shown below. Now we have created 2 data frames p d and p d new.

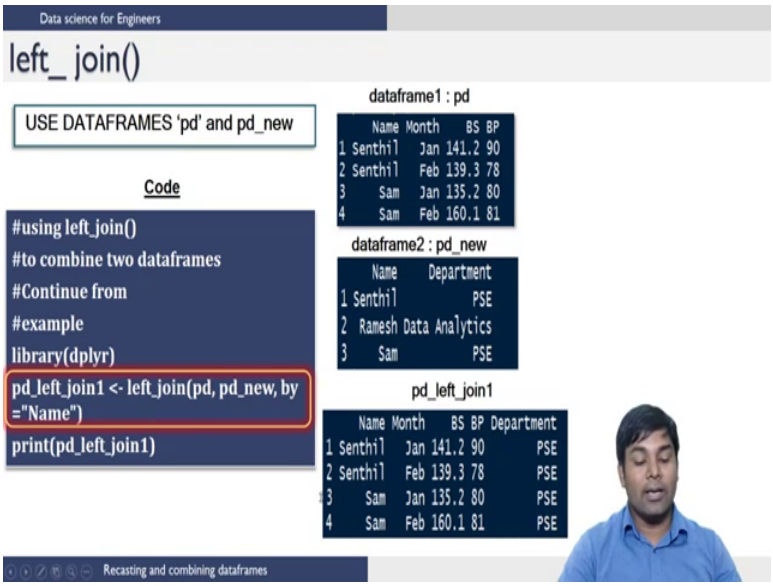
(Refer Slide Time: 14:01)



Let us look at how left join works. When you want to combine this 2 data frames p d and p d new a left join joins, matching rows of data frame 2 to the data from 1 based on the I d variables. From the syntax we can see that function data frame 1 and data frame 2 is the syntax we have and you have to specify the I d variable as the last argument. So now, if you want to left join data frame 1 which is p d and data frame 2 which is p d new, what it takes as a reference is, the data frame 1 which is p d and now it matches the rows of the data frame 2, which is Senthil Ramesh and Sam and sees in the data frame 2, what is matching with the name variables in the data frame 1, essentially it will keep only Senthil and Sam and not keep Ramesh, because it will take to matching rows from the data frame 2 to the data frame 1.

So, well see that when you do the example, now here there are only 2 I ds corresponding to the values in p d new and that will be merged with p d, the variable department will be added to the final data frame, only for Senthil and Sam.

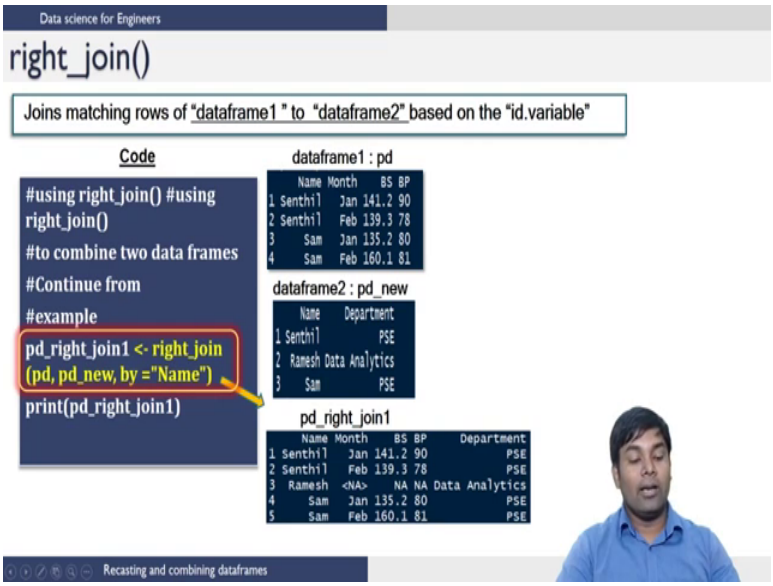
(Refer Slide Time: 15:23)



Let us see in detail, you have 2 data frames you need to load the library dplyr and you are doing it, a left join and I am naming the new data frame, which is coming out with this left join operation as, p d underscore left underscore join 1. I have to use this command left join, this is the data frame 1 I am passing in and this is the data from 2 I am passing in and then I want to join this 2 data frames by the variable name .

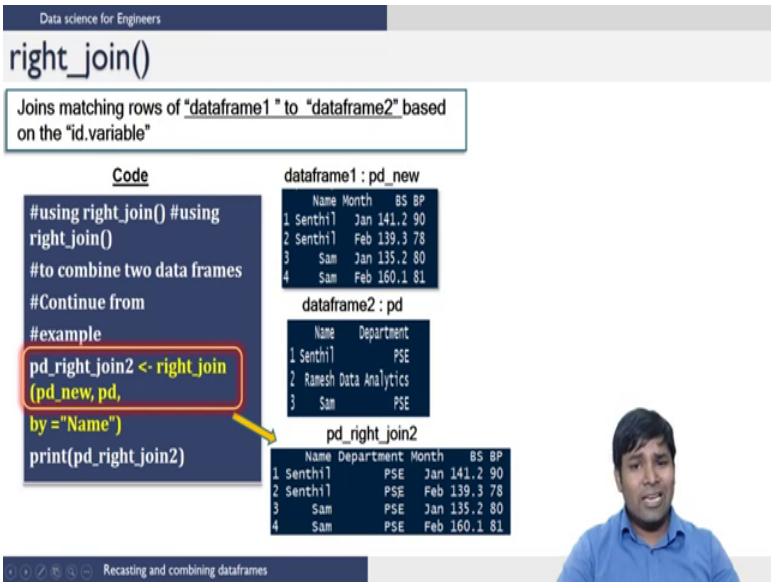
Now, when you specify I want to join this 2 data frames by name, the left join it will take p d as a reference, look for the names that are common in both p d and p d new and then take the data from the p d new, and merge it with the p d and then create another data frame, which is given by this name p d left join 1. So, you have data frame 1 p d which contains, Senthil and Sam and it look for, Senthil and Sam in the data frame 2 and then merges the information, that is available extra for these names and add it to the existing data frame, with another column department and the department of Senthil is PSC, in the department of Sam is also PSC, it will rehold the p d and then add the corresponding piece of information, that is coming from the data frame 2.

(Refer Slide Time: 17:00)



Now, let us look at the right join similarly. So, what right join does is, it is joins matching rows of data frame 1 to the data frame 2 based on the I d variable. Let us say, you have this data from 1 which is p d and data frame 2 which is p d new and you can do the right join, by using right join command and you need to pass what is data frame 1 and what is data frame 2, here we have p d as a data frame 1 and p d new as a data frame 2. Now, what is it take is it will take the p d new as the reference data frame and try to match the rows, which are present in the p d new and look for a match in the p d. We have Senthil and Sam there are matching in the p d also and it will keep this Ramesh now, because the references is this data frame. So, you will have Senthil Ramesh and Sam, but for Ramesh you do not have month, blood sugar and blood pressure values, which are replace by n s when the matching operation is that.

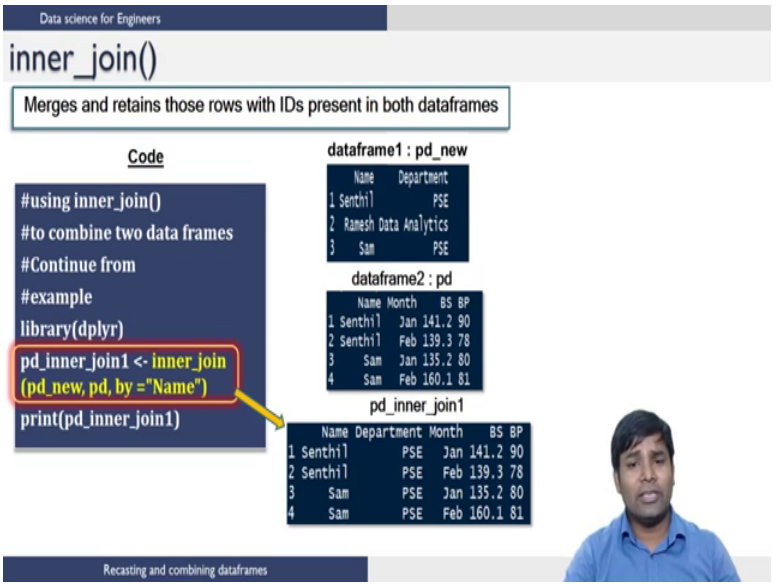
(Refer Slide Time: 18:10)



You can change the order, in which you pass the data frames and you can see that, if you change the order, you pass the data frame one has p d new and data frame 2 as p d. You can observe that output is similar to the left join, because now the reference variable here is p d, when you are using p d as a reference data frame even though you are doing this right join operation, the operation is similar to left join because your p d is the reference at the right join here.

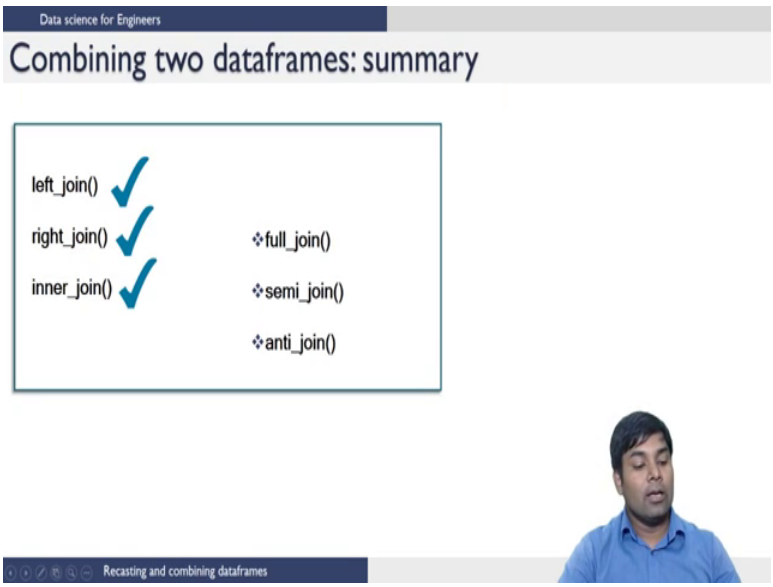
So, to summarize left join and right join can be used vice versa, but depending upon the way you pass this data frames, the matching operations will either look similar or different. So, you have to be careful when you are passing the arguments, to this left and right join commands.

(Refer Slide Time: 19:09)



Now, let us see what inner join does, inner join merges and retains those rows in the ids present in the both data frames. Now you have data frame 1 which is p d new you have data frame 2 which is p d, now when I pass these 2 data frames as an argument to this inner join function and I want to match them, by name it will look for the rows with I ds present in the both data frames. In this 2 data frames we have Senthil and Sam present, it will print only the data that is corresponding to the Senthil and Sam, because Ramesh is not available in this data frame 2.

(Refer Slide Time: 19:54)



So, we have seen left join, right join and inner join. We left as an exercise for the viewer, to understand how full join semi join and anti-works. To summarize in this lecture, we have seen how to recast the data frames? And how to combine 2 data frames using the dplyr package? In the next lecture we are going to see how to do arithmetic logical and matrix operations in r.

Thank you.