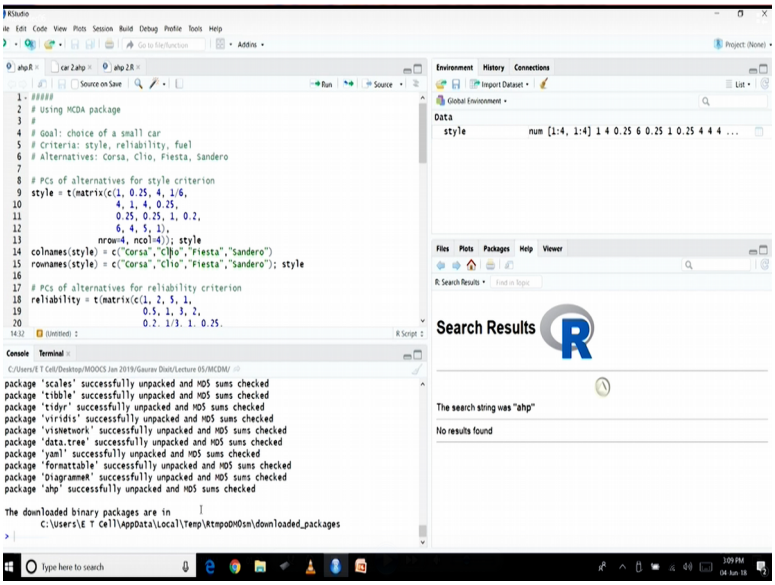
MCDM Techniques using R
Prof. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology – Roorkee

Lecture – 6
Analytic Hierarchy Process (AHP) – Part III

Welcome to the course MCDM Techniques using R. So in previous few lectures, we had been discussing AHP that is Analytic Hierarchy Process. So specifically in the previous lecture, we started our discussion of this particular example in RStudio environment that is choice of a small car. So let us continue from where we stopped, so let us go back to RStudio environment.

(Refer Slide Time: 00:59)



So in the previous lecture, we talked about the RStudio interface. We talked about installation of this particular set up, that is first we need to install R, then we need to install RStudio. Then, we also talked about this interface and different panels that you can see here and the importance of these panels. We also looked at if we need to install certain package, how to do it, so install.packages is the command and we need to go to this console section of this particular interface.

Any package if we know the name of the package, we can always go ahead and install it in this console section. **(Video Starts: 01:56)** So let us get started with our exercise and let us start execution of the code that we started discussing in the previous lecture. So as we discussed here, 3 levels of the structure that we talked about. Goal is choice of a small car and

as you can see here we have these 3 criteria; style, reliability and fuel. We have these 4 alternatives; Corsa, Fiesta, Clio and Sandero. So these are 4 small cars that we have as alternatives. So this is a typical example of a decision problem where there is just one decision maker.

Typically, this is problem related to family where the family is trying to decide which car to pick and typically one person is playing the role of decision maker. So as we talked about, we need to get pairwise comparisons from decision maker, so we need to construct comparison matrices. So first, we will start with comparison matrices for alternatives. So as we have talked about these pairwise comparisons of alternatives are to be with respect to immediate upper level that is criteria.

So in this case, we will first start with this style criterion. So, first we will construct the comparison matrix for alternatives with respect to style criterion. So here, you can see the code here style and we have these 2 functions to create the matrix. So you can always go into the help section and we can always type matrix to find out what this function actually does. So you can see here, so this is how we can always use the help section here in this panel.

So this is part of the base package that we have in R and this is the matrix function. It creates a matrix from the given set of values. So, specific arguments of these function you can see here in the help section. So you can see data is the first argument, then number of rows and number of columns, and there are certain other arguments also there. So typically, the important arguments are data, nrow and ncol. Others are default values, typically they are fine.

So you can see here in the script section that because we want to create a 4/4 matrix because we have 4 alternatives. So these 4 alternatives are to be compared with each other. So we are going to create a 4 x 4 matrix, so therefore we need 16 values. So in the data argument, we need a vector of 16 values, so that is to be entered and then number of row it is to be 4 and ncol is assigned as 4. so you can see here in this code nrow and ncol, they are 4, and you can see that 16 values have been entered.

So these 16 values are nothing but pairwise comparisons of each alternative with the other alternatives with respect to this style criterion. So another function that is being used here is t,

so let us understand what this function does. So let us type t and you can see this is the transpose function. So given a matrix, it returns the transpose of that particular matrix. So let us compute this code. So once this matrix is created and converted into a transpose, this is going to be assigned into this style.

This semicolon that you see, this is after the semicolon we get the next instruction which is nothing but we are trying to access the output of a style. So, let us execute this line. So you can see here 4/4 matrix in the console panel and you can see that 16 values had been displayed here. Let us execute next two lines to get the appropriate names for these columns for the matrix and the rows of the matrix. Now you can see here in the column we have the name of these 4 alternatives and in the rows also we have name of these 4 alternatives.

So in the matrix, we can see that each alternative has been compared with the other alternatives. Because this is 4/4 matrix, the alternative is also compared with itself and that value is 1, so that is there, and then it has been compared with other alternatives as well, so those values are given here. So 0.25 is actually when we go for seeking responses from decision maker, we would actually be getting 1/4, so which is mentioned as 0.25 here.

Similarly this value 0.16 is actually I know 1/6 is the response that we will get and it has been indicated in the decimal format here. So these are the PCs, pairwise comparisons, of these alternatives. So this was with respect to a style criterion. Similarly, we need to construct other comparison matrices with respect to remaining criteria. So next one is reliability, similarly we will construct this one. So let us run this code. The code is similar as we saw for the style. Let us go through this. So you can see the output here again.

With respect to reliability criterion, this is the comparison matrix we can see here, and again these 4 alternatives, these 4 small cars, have been compared with each other and the appropriate values as given the by decision maker can be seen here in this matrix. Similarly move forward, now with respect to the third criterion that is fuel, we can again construct a comparison matrix. So again let us execute this code and column names row names and you can see again the similar kind of comparison matrix here.

So now for 3 you criteria that we have that is style, reliability and fuel, we have constructed 3 comparison matrices. Now, we need another comparison matrix for the criteria, so before we

actually construct the comparison matrix for the criteria, let us combine the results of these 3 comparison matrices for alternatives, they are for the alternatives, so we would like to have a look of all these comparisons in one go, then we can use this command list.

So that will create a list data structure based on these 3 matrices that we have just now computed. So let us execute this code and in the console section in the output you can see a list of 3 matrices has been created. So the first element of this list structure is style that was for the style criteria and the matrix is there, then for reliability and fuel. So within one data structure, we have combined these 3 matrices. More information on list how this function actually works, you can always go to the help list and type list and find more details about it.

Now let us come back to our fourth comparison matrix that is for criteria and these criteria are to be compared with respect to goal. So since we have 3 criteria that is style, reliability and fuel, therefore we need to compute a 3/3 matrix. So in the in the argument section of matrix section, you can see nrow is 3 and ncol is 3 because we want to compute 3/3 matrix and therefore we just need 9 values here and these 9 values have been mentioned in this know matrix. So let us execute this code and we would get this matrix.

So let us create the column names, row names, and this is the 3/3 matrix for criteria and you can see in the row side we have style, reliability, fuel and in the column side we have style, reliability, and fuel and 9 values we can see here. Just like the other matrices that we have created 0.5 is actually 1/2 the way we take the responses from decision maker, 0.33 this is actually 1/3, 0.25 is actually 1/4. In the sense the way responses are taken here, they are indicated in the decimal format.

Now as we talked about in the previous lecture, one important aspect of AHP is that we need to check the consistency, these comparison matrices whether they are having the adequate level of consistency or not. So how do we check that. So first we need to load this library MCDA. You remember that this is the package that we have installed in the previous lecture. Now we need to load this library here in this particular instance of execution done that we are doing, so that we are able to access the function for execution.

So we need to load this library. So let us run this code. So this library is now loaded. Now all the function that are part of this particular package we would be able to use them. So the

function that we are going to use for the consistency check is actually the pairwiseConsistencyMeasures, this is the name of the function, and more details about this function you can always find out from the help section. You just need to type the name of the function and you will get the manual page for it.

Now you would see that we are using a $ notation here and then CR. CR is the consistency ratio the matrix that we are going to use for our analysis. At this point, I would also like to tell you that if you are not familiar with the R language and the syntax that is part of this code that we are executing, you can always go back and refer to the relevant lecture in my previous course Business Analytics and Data Mining Modeling using R.

There are lectures where I covered introduction to R so that you can use those video lecture to familiarize yourself with the R syntax and there you would understand what we mean by these notation $ and other things. It would be more comfortable for you to go through this execution that we are doing right now. So let us compute this, let us execute this line of code. So we get this value 0.16 which is equivalent of 16%.

So as we talked about, it should be less than 10%, so this seems to be on the higher side, therefore this particular comparison matrix should be reconsidered. Let us look at other comparison matrices, 0.07, so this is within limits. For fuel also this is within limit 0.08, so less than 10%. Then for the criteria also this is much less actually. So we can see out of 4 comparison matrices that we have here 3 are okay, only first one for the style that seems to be having more than allowed limit. So in this case, it is 16%, allowed limit is 10%.

However, at this point of time, we like to ignore this. Otherwise, the next step that is to be done is to reconsider this comparison matrix and find out inconsistencies and correct them and then again get back. Now as we said since we are using MCDA package, so in this the main important function that we are going to use for our AHP is actually AHP function you can see here, so this is part of this MCDA package.

So this is the name of the function that does the main computations for us. So we are calling this function AHP and the first argument is criteria PC that is criteria pairwise comparisons and then the second argument is alternatives PC that is having the pairwise comparisons for

alternatives, and once we call this function, so I will be storing the AHP scores in this overall vector that we have here. So let us execute this line of code and you would see.

So you can see here AHP codes have been computed, so it is comes out to be 0.29 for this car, then 0.27 for the next car, then 0.69 for the third car and 0.36 for the fourth car. So we just look at the AHP scores, we can clearly see that it is the fourth car Sandero which has been given the highest AHP scores given the criteria and given the alternatives that we have and the pairwise comparisons that were provided. So it seems that Sandero is the car that we would like to buy.

If you want to indicate the ranking in the clear sense, then you can use the rank function and we can pass on this variable that we have just computed overall and we will get the ranking. So you can see here Sandero is ranked one, then Corsa, then Clio and then Fiesta. So these are the 4 cars. If you want to see a graphical depiction of this ranking, then this is the function plot, alternatives, values, preorder that will give us this graphical depiction.

However, right now, this function in the R version that we are running is not supported, so if we run this code, we will get an error, you can see, Rgraphviz package could not be loaded, so there is this error. However, now after running this function, we can see that which alternative is finally selected. Now this was when we used the MCDA package, so every package and the kind of functionality that is provided as part of the package can be limited in some sense.

For example if we are interested in looking at the weights for each criteria, what were the weights which were computed, so they have not been displayed, they cannot be displayed using this particular AHP function and this package. So that is one limitation that we can see for this particular package. So one important aspect of modeling in R is that we need to identify the relevant package, most appropriate package as per our requirements.

So we need to find out what are the packages which are available, so list of packages which have the implementation of a particular technique, so in this case AHP, and see which packages are giving us the information that we require and are more relevant for our purposes. So there is another package that is available which also has the AHP

implementation, rather it is specifically devoted to AHP, and the name of the package is AHP itself and in the previous lecture itself, we have gone through the installation of that package.

So what we will do is now the same exercise choice of a small car that we have gone through right now. We will go through it again using this second package that is AHP. So let us go to the next file that we have here ahp2.R. So you can see the first comment that you can see in this script is using ahp package. So, this particular code has been written in that. However, this particular package requires us to specify our model in a specific particular file format which is called .ahp.

So before we can even move ahead, we need to specify our ahp model in the prescribed file format for this package. So, we will go to this particular file car2.ahp which is actually written in that format. So, I have already written this file for this example choice of a small car. So there is a particular syntax. How this particular file or the ahp model is to be specified in this, so we need to understand the syntax and how to go about solving our decision problems using this package.

So first, I will take you to the help section to make you go through the relevant documentation that is available for this package. So we just type ahp here and in the search results, we would see a lot many things related to ahp, you can see here. So you can see most of lines of these results they are related to the AHP packages and you would see only the last line here, this is related to MCDA package that we have already used.

The other R related to the AHP package which are which we are about to use. You can see here ahp file format. Before we even start using this package as I said, we need to understand this file format. So if you click on this file format, you can see here more details about this format can be seen here in this **(()) (22:31)** space. You can see introduction, this describes the details of the ahp file format as well as the basics of the calculate process that is part of this package. So calculate is the function that is part of this AHP package.

So if you want to use this package and the function that are part of this package, you need to understand how this format is to be written, ahp file format is to be written. More details about this format is written here, for example they call it YAML. The ahp files are in YAML format and what is YAML, so basic points are given here. This is based on indentation. The

syntax is actually based on this. Comments lines begin with this. Multiple lines are indicated by this.

The few details about this format on how this is to be written are here. They have given the main elements of this format that you can see here. You can see version, we need to specify the version of the package, then the alternatives and the format that how they are supposed to be written. Then we have the goal in the second component. Then we have the children here. So within goal, we have to specify our criteria and sub-criteria also. So, this is the format that is to be used and then they have given the examples also.

They have described each of these components, alternatives, goal and criteria how they are to be written under this file format. As you scroll down this manual page, you would see examples also, so they have a vacation example that you can refer for more help, however, we will go through. This is the vacation example that they have where they have used the format and specified their model for the decision problem that they had.

You can always refer to this help section to understand more about this particular package and the function which are part of this package and the file format also. So what we will do is this example that we have discussed till now, the choice of a small car, I have already written the .ahp file, used the file format for this particular example. So we will go through this. So as you can see version 2.0 that I have written. Then, we start with the alternative section.

You can see it has been commented out and we need to mention the name of the alternatives that we have. So here these are the 4 alternatives that we have. So the particular syntax you have to follow as is exactly and these alternatives I have written here. Now, more details about these alternatives, so these alternatives are carrying certain attributes and you would like to use them, those attributes in your modeling process, for that you can always refer the help section.

Now after alternatives, we come to the next section that is goal section. So here, we need to specify our preferences. So first for the criteria that we have, so 3 criteria that we talked about for this particular decision problem is style, reliability and fuel. So you can see for these criteria, the pairwise comparisons have been appropriately mentioned here. You can notice the syntax, the syntax has to be followed as is, so you can see here.

Then in the next subsection here children, so under children with respect to every criterion, so style and then it would be followed by reliability here, you can see reliability here, style here. So within this, we need to specify our preferences through pairwise comparisons for all the alternatives. So you can see we have 4 alternatives so therefore these many comparisons have to be specifically written on this file format using the syntax that you are seeing right here similarly for other criteria.

So once this is specified, this model specification is complete. So first you need to create this file, for example we have created car2.ahp. So this file car2.ahp will have all the pairwise comparisons, name of the alternatives and all those details. So, all the details about the model specification, you can see here and once this file is created, we can always go back to our script and now we are going to use this particular file in billing and executing our model. So let us start.

So first, we will load this library ahp. So once this is loaded, then now we need to find out the files. So we have written car2.ahp, so this is the directory that we have. So we want to list all the files with .ahp extension. So you can see the second argument is about specifying extension, and full.names, we will get full names of those files in this particular directory. So this list.files is going to list us the names of the files there; however, this particular directory name I need to change because this was for a different computer, so this is to be changed here.

So, I will change this and name of the directory we need to identify here so you can see and we will go back. So in this list.files, first argument I need to specify the path of the directory where the files are located. So first, I need to copy this path and this I can specify here, so this is done. Now once this has been specified, I need to change it to R syntax as per R syntax. So all these backslashes have to be converted into forward slashes, only then it would be understood by the R syntax.

So this particular example that this particular typing that I am doing right now, this will also give you an idea about the importance of syntax in a particular programming language. So once this is done, we can execute. Now you can see there are two files, car2.ahp and car.ahp. So these two files are .ahp extension and they had been written in the file format required by

the AHP package. So we are interested in the first file that is there car2.ahp, so we would like to select this file.

So again using the same command, now we will like to record the path of this file, you can see here. By typing 1 here, I am selecting the first output of this command that we ran. So let us execute this code and the path is recorded now. Load function is part of this package which can be used to again load the file. So, let us execute this. There is some error, so let me go back again. So ahp file this we just created, so there was some problem it seems there. So this path we need to change here again.

So this thing that we copied there needs to be used here as well. So once this is done, we will get the output. So, let us execute again. Now if I run this again, then you can see name of the file recorded in this variable and we will load this. So you can see now this is loaded. You can also see that in the environment section once I execute something and that output is recorded in a particular variable, immediately it is also displayed in the environment section as well. So that is always you can refer back to this particular panel.

Now let us calculate this. So the calculate function will calculate ahp score, weight contribution and consistency ratio for each criterion. Once this is done, there is another function that is available to us called visualize. So if we execute this part of code, we will see a hierarchical structure the tree, you can see here in this viewer panel. So if we zoom into it, you would see that this is the hierarchical structure for our decision problem.

First we have root which is actually referring to the goal that we should have, then the criteria and then the alternatives. So we have 4 alternatives here. So this is the structure that we can create using the visualize function. Now since we have already done the computation, let us look at the results and analyze them. So we can use this function analyze and we will see the results. So you can see here in the first row, these are the values, ahp scores.

So if we go back and compare it with the results that we had got with the previous package, you will see though these results are expressed in the percentage format and the previous package we got the results in the in the decimal format, but the results are same. For example Sandero we have got 36.1% and you can also memorize the numbers for other alternatives. If

we scroll back, scroll towards the previous results, you can see here Sandero this was 0.36, so the values are same.

With the help of both the packages, we go the same output, so let us go back. Now, however, this particular package provides us more detailed output, you can see here. **(Video Ends: 34:38)**. So what we will do is, at this point, we will stop here and we will continue our discussion on these results that we have got and we will continue to analyze and discuss it in the next lecture. Thank you.