Applied Predictive Modeling

Lab 1, Part I Transcript: Introduction to R - Data Types

Hello everyone and welcome to ADS 503, Applied Predictive Modeling, Module 1 lab session. In this lab session, we will start with a basic introduction to R data types and other useful functions in our ecosystem. Then we'll cover some useful R functions from module one including Box-Cox transformations. Finally, we will cover PCA, also called the Principal Component Analysis. And this is probably the most useful technique that you'll learn from module one. So let's get started. I'll be using a R Markdown document for this lab session, as you can see here. You should have access to this document on the course web page, and R Markdown documents are very similar to Python Jupyter notebooks.

If you have not looked at them before, each R Markdown document will have what you can consider them as chunks, so, anything that starts with this, R datatypes here is a chunk and then as you go down, you'll see multiple chunks. So you just divide your document into chunks, so that each independent part is in a chunk and then you can create a document that explains your analysis as a sequence of steps.

Let's get started with the R data types. So for those of you who are not familiar with R, R has many data types, and I'll review few of those, specifically, vectors, matrices, data frames and list data types. And this document also has a link, right here, that you could use to dig deeper into other data types or even the data types that I'm gonna mention. So in R, there are two ways you can assign something to a variable.

One is this operator and the another is the equal to operator. So, so here, I'm assigning xn, this numeric vector c(1, 2, 3). So let's see what happens when I do this.

You can see that, on the right side you can see that, it has created a variable xn which is a numeric and it has values 1, 2, 3. So if I do the same thing with yn, you can see that xn and yn are identical, with the same numeric 1, 2, 3's.

These are two different ways you can assign variables some values associated with them. And another very useful function in R is called the str function.

You can pass an object, R object to this str and it will give you the structure of the R object. So if I click this one, and then if I look at what's in here, so you can see it's a numeric 1, 2, 3, which is what you see on the right side here too in the environment. If you look at the structure of other objects you'll find more useful information but this is a very simple numeric object. So that's why you just see numeric 1, 2, 3. And then another useful function within R is the class.

If you pass an object to the class function, here again, I did the same thing, and then you can see, it says it is a numeric object, which is what this object is as you know from here, it's a numeric vector. So like numeric vector, there are character vectors.

I can create another vector xc, here instead of it's numeric, now we can clearly see it's a character vector which has three values, a, b, c. And then again, if I do same thing, structure , you can see it's a character vector. It has values, a, b, c. When I do class, it's going to tell you it's a character vector. Those are the numeric and character vectors. So the other important data type in R is the matrix.

Here, what I'm doing is I'm creating a matrix. So this is the function, matrix, and so if I hover it, it tells you, "Hey, it requires a data and the number of rows and the number of columns". And then there is this argument called byrow, which we'll learn about in a few moments. So here, I'm just saying, hey, create a matrix with five rows and four columns and use this values to create those 1 to 20.

This is a sequence from 1 to 20. So let's see what happens when I create that. Here we can see xm, it's a five, five rows, four columns, and then it has these values. And then if you want to see what's in xm, so when I press xm, you can see in the console that you have four rows or five rows and four columns, and the values are assigned column-wise here. So, the first five columns, or the first five values go to the first column and the next five values go to the second column, and then subsequent five values go to the third column, and the last five go to the fourth column.

How matrix is defined and what are the options that are available, can be found out by using this command, if you do question mark in the name of the function, you usually get in the help box what that function is, what are the arguments associated with it, and then some examples. So that's the best way to learn about functions in R.

Here we observed that, hey, one to five went to the first column, but maybe you want the columns to go or the values to go row wise rather than the column wise. One thing we can do is, to do that is you can say, hey, fill these values by row. By default, if you look at the matrix here, the byrow is set to false. If we set byrow to true, then we can make it go across the rows. Let's see what happens if we choose that option. So here you have new matrix, xm1.

When I click it, you can clearly see now it has started filling out the rows first instead of the columns first. It'll start filling the row, first four values, go to the first row, next four go to the second and so on. Again, you can call the same str function to get more information about this object, and then you can call the class function to get more information or what class it belongs to, which is matrix in this case.