Practical
Spatial Statistics and Spatial Econometrics
With R
Prof. Saif Ali
Department of Social Sciences and Humanities
Indraprastha Institute of Information Technology, Delhi

Lecture - 48
Session 2

Hi, welcome to another practical session. My name is Saif Ali, and we are learning how to do Spatial Statistics and Spatial Econometrics with R. Today's topic is how to work with Tabular data in R. It is a fairly basic topic. We are not going to touch on a lot of spatial concepts, but it is something that you need to know in the beginning as to how to work with data before you can start doing other things.

Before we get into the topic today let us just recap briefly we said that excellence and mastery over this subject or any other subject is achieved by a combination of understanding and skill. And, understanding is gained by listening, reading, thinking, solving things on your own putting pen to paper.

Whereas, skill is gained by applying that understanding to real-world problems. The way to gain skill is by actually doing something, doing it yourself, trying it out, failing trying again, and keep going and writing a lot of code. So, in other words, the wrong way, the way not to acquire a skill is by watching a lot of lectures. So, for example, if you just watch this lecture, and read a lot of programming books and forums you might acquire a lot of information, but you are not going to acquire skills.

So, I will just reiterate, they encourage you to follow along as I write code you should go to your own R studio and try out the same code and come back to the video. That is the best way to learn. That is the way it is going to be fun for you.

So, with that said what are we going to do in this session and what should we know before we go into this session, let us just talk about that briefly. So, what should you know? In terms of understanding, by now you should know what R is, what is RStudio and what is an R package. This is something that we spoke about just R fundamentals. You should be familiar with the tabular structure of data.

So, the tabular structure is when data is organized as a table in rows and columns. The most common example is Excel, if you worked with Microsoft Excel or Google Sheets, a spreadsheet is the classic example of tabular data. And, also it will be helpful to have a basic understanding of programming fundamentals like you should know what a data type is.

For example, a numeric data type that stores numbers is different from a literal data type that stores letters and symbols, and characters. It would be good to know things like a variable like what is a variable in a programming language and what is an operator.

So, examples of operators are plus, minus, division, and assignment operators. So, these are programming fundamentals. I think all of you will be familiar with them, if you are not I will do my best to explain as I go. If you are completely new to programming, then I do recommend along with this material quickly reviewing some basic fundamental programming either from a book or from your favorite videos online. It should not take long.

And, you should know what the CSV file format is. The CSV file format is a standard file format for storing and transporting tabular data and this is a file format that is understandable by Excel, Google Sheets, and also by R. So, if you understand this format you can send data around in a bunch of these programs.

Now, in terms of skill, what should you have already done? Well, to go into this lecture you should already have set up your computer for development with R. If you have not done that obviously, you are not going to be able to follow along with me. You are not going to be able to write code.

So, please do that before going further in this video. It would be a good time to pause right now and set up your computer for R development and if you want to know how to do that go back to the previous video in which I show you how to do that. So, for this video, I am assuming that you are ready to go. You have a window open right now, R studio window open along with this window that you can just quickly tap to try out code.

Also, you should have installed the gstat package this is also something that we did last time. And, with that, what will we do now? We will install another package called sp and the sp package is a package that provides all kinds of methods for spatial data analysis. Then we will take some data and examine it using R view it looks at its attributes and properties and

then we will try and export it out to a CSV file because once you are done working with R typically you do want to write data out as a CSV file.

So, you can send it to other people or look at it using another program. We will do some basic stats, not spatial tasks just basic stats on tabular data. And, so, this is exciting. We are going to write our very first R script, for some of you this may be the very first R script that you have ever written. So, I hope that you watch till the end and we are also going to make our very first R plot.

So, you will have at the end of this video, a script that you have written and a plot that you have made using R. So, you would have made some progress. So, if that is clear we are going to go ahead and if you are missing out on any of the prerequisites please pause now, go back, and finish those.

So, let us transition to our RStudio and, do some live code.

In the previous video, I talked about RStudio and the various components of the interface. So, therefore, you should be familiar with what we are looking at now. So, I have the console here. I have a R script that I have prewritten that we will go through and I will explain this and I have already created some data. And, these are my files and if I just tap on this one, this is where I will see the plot that I make. So, this should be familiar, I hope.

So, let us get into the code. So, at first, you will notice the color coding. The different lines of code are in different colors. So, any line that starts with a hash is called a comment. So, if you come if you want to write something inside an R script that you do not want R to understand or try and decode it is just a comment, it is just a note you put a hash at the beginning of that line.

So, for example, we had installed some packages last time. So, we had installed, remember the gstat package. So, I do not need to run this line of code. So, if I do not want to run something I just comment it out. So, if I remove this comment now it is going to run the code, but I do not want this. I have already done this. So, I am going to get rid of this and I am going to try and install as the first step the sp package. So, I am going to hit Save.

So, if you want to run this line of code, line number nine, then what you do is click on this button called Run.

So, if you click this, it is going to try and restart R.

And, now it is trying to restart. So, what did we get here? So, it says package sp is in use and will not be installed. So, I am going to go over here and check if it already has sp. So, it already has sp and that is why when I try to install it, it said the package already is in use and it is not going to.

So, sometimes you are going to do something, you are going to see an error message. So, all of the error messages will appear in the console. So, you should know how to read error messages, decode them and understand what went wrong. So, what is happening now is that we already have the package. So, I am going to comment on this line as well because we do not need it. And, now we are ready to load both the gstat and the sp libraries.

So, I should tell you that when you install a package it is a one-time affair. You only have to do it once. If you have one R installation you only have to install every package once ever, but you have to load the library every single time.

So, for these commands library, gstat, and library sp, you are going to have to run these every time.

So, I am just going to run these and whatever code you run, using this button you will see the results in the console. So, you should always be looking at the console to see what happens. So, it seems that it loaded the libraries, alright. And, the next line of code is interesting it starts with a question mark.

So, this is called an operator, an operator is a special symbol that has some special meaning to R, and in R a question mark followed by a package name means that you want R to open the help manual for that package. So, we want to open the help manual for gstat.

So, if we Run this, it will open the Help manual in this viewer pane here on the bottom right and this is the R documentation for the gstat package. So, maybe if I might ask you to do a small exercise here, how would you open the help manual for the sp package?

You can pause the video right now and try that out. I am going to show you how to do it. It is the same thing, question mark sp.

And, now in the viewer window, you have the help manual for the sp package. You can go through this and read this and we will explore these manuals more as we go along. If you want to clear the console you can press Control L and it will clear the results in the console. So far so good.

Now, what we want to do is we want to start looking at some tabular data. So, the sp package comes with some data already, it's part of the package. Usually, if you want to look at tabular data you have to read some data from outside. I will show you how to do that a little later, maybe later on in this course, but for now, we will just use data that comes along with the sp package.

And, this data set is called meuse and this is a data set that it is from some researchers Burrough and McDonnell from 1998 it contains data about heavy metal concentrations along the riverbed of a river called meuse. So, now, we do not know anything about this data. We do not know what it contains, or what kind of variables it has. So, we want to now start exploring and examining this data.

So, let us go ahead and load it with this command data meuse. So, once you load something it is going to show up in your environment. So, this meuse variable has shown up. So, I already had it. So, let us say if you want to remove some data you can just say remove meuse. So, it will get rid of. So, now, see it's gotten rid of meuse and if I run this again it will bring it back.

So, any data that you have currently loaded in your environment will show up here. And, now we want to see well what is inside. Let us start opening this up. So, this command class meuse, tells you what is the type of the meuse variable. So, meuse here is the name of a variable, it is a variable that we have created inside the R environment.

And, every variable has a type like I said before some variables are numeric in nature. So, they only store numbers some are literal and they store strings. So, what is the type of this variable we can know the type of any variable in R using the class command.

So, if we run this it tells you that meuse is a data frame and a data frame is another word for a table. It means that it is a table; it is a frame consisting of rows and columns and, every row is an observation and every column is a variable. So, if you look at the data frame in the environment window here, you can see that it has 155 observations of 14 variables. So, there are 14 variables and they have been observed 155 times.

So, now we want to see inside. So, we know the dimensions, we know the type, and we want to see inside.

So, one thing you can do is just click on this. If you click on this it will show you the whole data inside this sub-window here and you can see that there are variables called x and y. So, it seems like a location, there is something called cadmium, copper, lead, zinc, elevation distance, and so on.

So, I do not know what all of these mean. Usually, anytime you work with some data it will come with metadata; metadata means data about data. So, you should have some source of metadata that helps you make sense of these, so you do not have to guess.

And, in this case, the metadata for this data set, I have given the link here in this file. So, if you open this pdf, this tells you in detail about what each of these columns means. Another way to examine data is by using the head command. So, if you do head meuse what it does is, it just shows you the first few rows, I mean this shows you all of the data.

So, we know that it has 155 rows, but if you just want to look at the first few rows just get a preview you can use the head command, and you can also use call names. So, call names just tells you the names of the columns in the data set. So, by now, we know the names x y, cadmium, copper etcetera. It is the same names that appear in this row, right here at the top row.

You can look at the dimensions and before I do this you might pause the video and think to yourself what the dimensions are, we already know them it is 155 by 14. So, a data frame is like a matrix. It has 155 rows and 14 columns and each row is an observation and each column is a variable.

You can also look at individual elements of the data frame. So, this is a new operator. If you put numbers inside these square brackets, it tells you a particular observation, a particular element inside that data frame. So, we just want the first element the first row, and the first column.

So, if we Run this, it tells us that the value of this element is 181072. If you open this up, you will see that the first element here, the first variable, and the first row, it is the same value. So,

try using these square brackets to examine other elements in this data. You can also examine values from a specific column.

So, suppose, I am an analyst that only cares about zinc content concentrations. I do not want any of this other stuff. So, I just want to look at this column. So, what I can do is I can write the name of the variable and this is saying that I want the first row and I only want the first row from the zinc column and not anything else.

So, if I run this it tells me that the value is 1022. Again, if you look at the first value inside the zinc column it's 1022. So, these are all just fun ways to look at the data. You can also access individual columns using the dollar operator. So, if I run this it tells me, it prints out the whole zinc column.

And, if I just want the first few values I do head. So, I am combining the head function with the dollar operator to look at the first few values inside the zinc column. So, I am just going through this line by line, you are most likely not going to remember it if you just listen to it. So, you can stop now and try to run this code, get till here by yourself, before we go further.

Welcome back. Now the one thing to know is that a data frame is a composite data type. It has many different types of data structured as a table, but each column within that data type, within a data frame must have a single type. So, for example, in the zinc column, if we print out the class, the type of that column, we are told that it is a numeric type. So, all of the values in a particular column of a data frame have to be of the same type, you cannot mix.

So, if zinc is a numeric data type then all of the values have to be numbers. You can have different columns that are of different types, but within the same column, you cannot have different types. So, that is something to keep in mind. Similarly, if we print out the type of land use column, then we are told that this is something called a factor.

You do not need to know what a factor is. All you need to know is that the land use column inside the meuse data, which is this. It seems to have some letters, which is a different type from the zinc column. So, you can have different types in different columns, but within the same column, all of the data has to be of the same type.

So, how about if we have this data meuse inside R, but what if we want to export it, we want to look at it in Microsoft Excel or we want to upload it to Google Drive or we want to send it

to somebody else? How are we going to do that? Well, you can use a function called write dot CSV and then if you write this data out it writes the first parameter of this function is the name of the data frame that you want to write and the second is the paths to where you want to write it.

So, if we look at our data folder, we have a CSV called meuse which will contain the same data that we just exported. So, you can use that to export tabular data out of R and then share it with others or look at it in some other software. So far, so good. So, that was about examining data, but what if we want to do some statistics?

So, let us print out some summary statistics for the zinc concentrations. So, the zinc column contains concentrations of zinc along the riverbed. So, they measured concentrations of zinc at various locations along the riverbed and that is the data that we want and this is in parts per million.

So, we want to see what the summary statistics are. So, if you use this command called summary it summarizes the whole column and tells you what the minimum value is, the maximum, the mean. So, we know the mean is 469.7. So, on average the zinc concentration on the river bed is 469.7 parts per million.

Similarly, you can use the var command to calculate the variance or the standard deviation and finally, we want to plot the frequencies of this variable.

So, we can use a command called hist and if you want to look at any command you can type a question mark and type the command name and it would tell you how to use that command, it will bring up that Help manual. So, when you use hist you just need to provide the variable name of which you want the histogram.

And, if you run that it will show you the histogram of that variable in this window right here. So, these are the frequencies we see that values between 0 and 500 are very frequent and you know then it kind of the larger the concentration becomes you have fewer and fewer values. So, it may be like an exponential distribution or something.

That is it. That is all the code that we will discuss today. I hope that you will feel more comfortable working with tabular data in R and I encourage you to pause the video again and try stuff out for yourself.

Alright, I hope you had fun trying things out and writing code. I hope you got a lot of errors and you tried again. Just to summarize what we did today, we installed a new package called sp, and we took some data from that package called meuse and we exported that data out. We used a variety of functions to examine what that data looks like. We learned some new operators, I have given you some useful links here to user manuals for the various packages that we used.

So, you should feel free to explore these, and once you have done that you will be ready for our next practical session and I will see you then.

Thank you so much.