**Learning R**

### **R for data science**

Selecting transcript lines in this section will navigate to timestamp in the video

- Every element of our professional lives and even our personal lives is being transformed by data. How you find jobs, how you find clients, how you organize your time and choose projects, how you decide what has greatest value and the most potential for helping you achieve your goals, all of these are different now because of the ready availability of data. But if you want to get the benefits of this data revolution then you need to know how to work with data. And one of the best ways to do that is with R, a free and an open source language that was specifically developed for exploring and modeling data to help you find the insight that you need. I'm Barton Poulson and in this course, we'll take a look at how you can get started with R. I'll show you how to install R, the R Studio environment and additional code packages that extend R's functionality. We'll see how to make data visualizations, how to wrangle data and to calculate descriptive statistics. We'll work with some powerful methods for analyzing associations in data and building statistical models to help you get insight. We'll also see how you can document and share your work with others, so they can get the same benefits of the data revolution. This is an introductory course, so you don't need to have experience with R or with computer programming. It's helpful but not critical to have some familiarity with the basic concepts of statistical analysis but either way, I'll explain concepts thoroughly as we go through the course. You'll see the power and the flexibility of R and how it can help you make the most of the data that's all around. And that can help you reach your goals more effectively and more efficiently. And so with that in mind, let's get started with learning R.

### **Using the exercise files**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Narrator] If you have access to the exercise files for this course then you can download them to your desktop as I have done here. When you open this exercise files folder you'll see there are three sub folders. The first one is Code and this contains the R scripts that we'll be using in this course. The second folder is Data and it contains two formats of the same data set called state data that I'll be using as example data. It also contains a code book that explains the variables and the sources for the data. There's a third folder called Output and this is empty right now because you can use it to save any images or other output that you create. In addition to these three folders there are two other files. There's a read me file which simply explains what's in this folder. And then there's an R project file which helps organize the data in the software that we'll be using in this course. If you don't have access to the exercise files that's okay. You can still follow along by watching how I use the course files and adapting it to the files that you have available on your computer.

### **R in context**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Instructor] When it comes to working with data you're confronted with a potentially overwhelming range of choices that are competing for your attention. The first and most obvious choice for working with data really should be a Spreadsheet. I like to think of these as the universal data container because really they're everywhere and everybody uses them all the time. Obviously, common choices for Spreadsheets include Microsoft Excel and Google Sheets, although there are many other possible choices. Spreadsheets are great because they let you organize the data however you want, they can sort and filter data, they can count and summarize values and they can quickly make graphs. Truthfully, spreadsheets are probably sufficient for the majority of real-world data tasks that don't involve creating models of your data. But when it comes time to move beyond summaries and basic graphs and start making those statistical models then you'll need something more specialized like a statistical application. Some of the most common statistical applications are SPSS and SAS and my personal favorite jamovi, the open-source application, all of which give user-friendly point-and-click interfaces for common tasks and data exploration and modeling. But you may have data that doesn't fit nicely into the rows and columns that standard statistical applications expect or you may have questions that go beyond what drop-down menus are we able to do. In that case, you'll need to take the final step to a data oriented programming language because that gives you the ultimate in control and power in analyzing your data in the ways that specifically address the questions that matter to you. Now, when it comes to analyzing data some of the most common and powerful choices for that kind of programming are Python, this is a general purpose programming language that's been very well adapted for use with data or Julia which is a small niche program, a newcomer in scientific computing or R, a language specifically developed for working with data and as you know the subject of this course. Now I want to start by pointing out where R fits in the statistical programming framework and I'm going to give you a few different perspectives on this. One is a very common survey called the KDnuggets Poll of data science and machine learning software and this is the most recent one from 2019 where they ask people what programs and languages they use on a regular basis in their work. And traditionally R has been at the top of this list but in the last few years Python has taken over the lead and this particular year, it was followed by a specialized application called RapidMiner although the KDnuggets people say this may have been the result of very active sharing of the survey among the RapidMiner community. Third place is R at 47%, about half the people said that they used R. I want to point out that's just right above Excel, the spreadsheet or general-purpose data container. So according to this one R is still in the top five. I have the top six here. There are many more choices. But R is a solid choice for working with data. For a different perspective on this we can look at the data science jobs that are listed on Indeed. And the most recent data for this comes from 2017 and again, Python is here at the top of the list but R is there at fifth on the list. Again, the list is much longer with about 13,000 jobs on Indeed posted asking specifically for expertise in R. In fact what's surprising, that's even more than for C, C++ and C#. And then for a third perspective on this one using data from 2018 is a tabulation of the number of scholarly articles where they mentioned that they used one particular language or program or another. And in this case, the point-and-click application SPSS is far and away the leader with over 80,000 articles. But R comes in a respectable second place with about 50,000 articles. Now what's interesting about this one is in this particular list Python doesn't even make the top six. And so, it lets you know depending on your perspective the different programs and languages come up in different rankings but R is always there and that's an important one. And I mentioned that because there's a very frequent but maybe not so productive debate in the data science world and that is Python or R. Meaning should you learn Python or R or which one is better or who wants to be on a team and throw rocks at the other person. These are two of the most common choices. But I think to a certain extent it's sort of like asking somebody whether they like Star Wars or Star Trek better. To outsiders, they can't even tell them apart. They have spaceships, they have lasers and seem to be same people. And so this may be a point of drawing a distinction without too much of a difference. But let me mention this. Python is very popular and Python has a couple of very good things going for it. Number one is very strong in machine learning. And also if you're developing an app that uses data, so a data-based app, then Python might be a very good choice. Python also has the strength of being a general-purpose programming language. You can do anything with Python and for people coming from computer science, this is a compelling choice. On the other hand, R is especially strong in data analysis. Think statistics and scientific research. But python can do statistical analysis, R can do machine learning. The overlap between the two is tremendous. And it makes me think a little bit about another comparison, say for instance, you're trying to promote your business online. You wouldn't ask whether you want to be on Facebook or Instagram. The answer of course, is you want to be on both and several others at the same time because they each have their strengths. They each help you reach slightly different people in slightly different ways even if there is a tremendous amount of overlap between them. And so, the same thing is true for a professional data scientist. If you're going to work in the field you're going to be expected to work in many different languages including R and Python. So there's a huge advantage to learning R which is what you're here for at this point. R is a great way to work with data. It's at the top of the list on almost any way you look at the usage of data. It has a great community supporting it and it'll get you started on your path to working with data, finding insight and getting your work going.

**Statistical Tool**

SPSS  
SAS  
JAMOVI

**Code language**

Python or Julia or R

Data analysis specialist R

* **Question 1 of 1** When would you use a data-oriented programming language, such as R?  
    
  + When you need to count and summarize values, and quickly create graphs.
  + When you have data fitting nicely into the rows and columns that standard statistical applications expect.
* When you need to analyze your data in ways that specifically address the questions important to you.  
    
   Correct

When you need to sort and filter data, and organize the data however you want.

### **Installing R**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Narrator] The first step to working with R is to actually get R on your computer. Fortunately, this is easy and it's free, that's a huge advantage over many other ways of analyzing data. To install R, go to the R website that's at r-project.org. And when you get to that homepage, simply click on this link that says download R. Now when you do that, it's going to give you a lot of different servers, mirrors, that you can click on to download R. You could try to pick one that's geographically close to you, but far and away, the easiest way is to simply click on this first one which will automatically pick the closest. Once you've done that, you simply need to click on the appropriate link for your operating system. For instance, I'm on a Mac, so I could click on this one right here, and then I just come down to this package. I click that and download it and install it like any other program you can do. If, on the other hand, you're on Windows, we'll click on that, and then here it very clearly says, if you're installing R for the first time, you want to install this base binary, so use that one. And if you're on a Linux system, well, it's a little more complicated, but if you're in Linux you probably know what to do. But any one of these you simply download the R package, you install it like any others, and then you are theoretically good to go. There is, however, one additional step. There are many different environments, or different programs or different methods for running R and that's what I want to show you in the next video.

### **Environments for R**

Selecting transcript lines in this section will navigate to timestamp in the video

- When you actually want to start working with your data in R, you have a lot of choices on how you interact with the language. Some of them are more common, some of them are easier, some of them are more accessible. I want to show you a few of the common options and you can choose one that works best for you. The first one is to actually use the R application that you just installed a moment ago. When you open it, what you're going to see is this window, the R console all by itself. Now you can open up a separate scripting window and save your commands and I ran these three commands one at a time. They show up in the console and the last one produces a plot that's in it's own separate window here. Now this is one way to do R. On the other hand, it's not necessarily the easiest. If you like working with text commands, you actually do have another interesting option and that's to use the terminal. In a Mac, you simply open that up and once you've installed R, you can simply type the capital R and you get the boiler plate text and you can start doing the same things that you've done elsewhere. I can type in plot. Iris that's a dataset that's commonly used for demonstrations. I run that command and look, I get the same chart. This is another way of doing it. If you're coming from Python, then you know that one common way of doing data analysis in Python is with what's called a Jupyter notebook and you can use those for R as well if you're using continuum Anaconda's installation of Python, there is an option to use R within that. It's a little tricky to set up. An easier way to do it is with something like Microsoft Azure notebooks. This is Jupyter and it lets you do a number of languages including Python and R in an entirely online format. You create a free account and then when you sign in, you'll get something that looks like this. It's a Jupyter notebook and so for instance, I can run these commands and then I can actually share the information, save it and it works really well. And if you're on something like a Chromebook, where it's difficult to install things locally, this can be a wonderful choice. However, in the R world, there's another framework that's more common it's something called Rstudio. And before I go away from that browser, I want you to know that there's an online version of Rstudio as well. You can go to Rstudio, but the easiest way is to simply go to Rstudio.cloud and again create a free account and when you open it up it looks like this. It's actually exactly what Rstudio looks like. It's just a little bit smaller because of the browser information around it. But this is a nice way to work. It's a unitary window. You put your commands over here. You get your console output here. You save the environment objects over here. You get your graphs down here. And we can zoom in on those so you can see them a little better. And so this is one way to do it, but far and away, the most common way of using R is in Rstudio on your desktop. And that's what I've got right here. And in the next video I'm going to show you how to download this and then we'll talk about navigating it and we'll do the rest of the examples in this course using Rstudio on a local installation. But of course you can do it with Rstudio cloud and you can adapt this to the Jupyter installations or the Jupyter online access or to the R app or even to the terminal. Use what works best for you and fits in best with your own workflow. On any of them you're going to get the power and the flexibility of R. And so let's move ahead and take a look at the Rstudio environment in particular.