Programming with R

by Mihaela Danci

Start Course

Bookmark Add to Channel Download Course

Course Overview

Course Overview

[Autogenerated] Hi, everyone. My name is my holiday change and welcome to my course programming with our I would data analyst. And let me tell you something about me. I'm really passionate about finding patterns and working with visual datum. Data analysis is an incredible skill in today's market, and in this course we're going to explore the most important concepts that you were required. When four starting to work with art tell off the major topics that will recover include discovering the syntax by working with our studio. Then we'll dive into data types and we will work with different data structures like factors or data frames. Next, using Contra Flow state. Once we will control the execution flow off our program and finally we'll create our first function. By the end of this course, you'll have a solid foundation to get started performing data analysis. This course is a quick introduction to our programming. So no prior experience with our is required from here. You should feel comfortable diving in tow are with courses on graphics, exploratory data analysis and data science. I hope you enjoy me on this journey to learn our with the programming with our course at plural site

Getting Started with R

Introduction

[Autogenerated] Hi, My name is Mihailo, and when I was a child, I was in love with Matt. Later, I discovered the data analysis field, and since then I spend my time exploring various algorithms to see what's behind simple numbers. In my spare time, I like to travel and to discover new people and new cultures. I am delighted that you've chosen to take the scores because ours popularity has increased in the past years. The amount of data that we create every day. It's simply shocking. In only one day, we watch millions off hours off video content and exchange billions off messages. This numbers are not going to decrease at all. Actually, we will create even more conduct so much that by 2025 it's estimated that each day globally content will be created equivalent to more than 200 millions. David Ease. Can you imagine that? Learning a new language to analyze data? It's one of the best things that you could do. Right now. There are many choices out there, like by turn our SQL and many more. When you decide to learn a programming language, you have to think about her purpose. What do you want to be able to do? What kind of analysis do you want to perform when learning are which, by the way, it's free and open souls. You'll discover the statistical world because our was written by statisticians for data analysis, you can important data from various sources, check your data, perform analyses and will powerful graphs. You can save all your work into scripts, so you can you use it later. We will focus on our as the programming language, learning basic things such as setting off the environment, discovering the syntax off our data types and the infrastructures. Then we will work with the simplest data structure vectors followed by matrices. Next, we will create an analyze the most used data structure data frames with control statements, as if an else will create our first function before jumping into the action. Let's dive into a bit off our history. Our is a programming language that is based on another one called S, which was created in 1976 at Bell Labs. The name are comes from his creators Rosie, Haka and Robert. Gentlemen who never loved are in 1993 The source code released to the public was a critical point in art history because thes transformative into a free software. In 1995 the first stable version was released to the public five years later, fast forwarding 11 years on integrated development environment called Our Studio was released. Our studio is an application that facilitates writing code in our language. We are going to use our studio for writing code alongside the latest version off our which at the time of recording this course is version 3.6 point. Oh, this is a beginner course. Oh, no. Preta quizzes are required during this module will take a look at how to install our windows. Don't worry if you are not a Windows user, are works fine with other platforms such as Mark or Lennox. With our environment set up will do some basic math calculation. Discovered assignment. Look at our data types of variable in our and the few fundamental rules related to the code style. Last but not least, we will create our first

project, using the latest version off our studio at the time. Off recording discourse, Let's get started

Why R?

[Autogenerated] you may be wondering why are so popular these days to show _____ power. We need a case that you're scarce will increase after each module, and we will help marry our character to serve several problems. Mary goes shopping at the supermarket called Global Mantex almost every day, because off this, it's hard. Two kids track off the product that she typically buys in the amount of money she spends in orderto do some analysis. Mary collected data regarding budget money, spent products, prices, quantities and if there were, like any offers or not these entire collection off data it's called a data set. We refer to this as the products data set. Many has a lot of questions to answer, such as. How much has she spending the last month, which is the most fragrant product bought, or what is the average cost of a shopping basket? Additionally, she would want to know what her profile is. Toe Farkas the products she will buy. And if there is a pattern in the offers in order to manage her budget better, how should she find the answers? What kindof tools should she use? American started friendly spreadsheets were the data. It's arranging columns and rows. She can filter and sword the data easily or even create some graphs. It is like a good option for small questions that she will most likely perform only ones toe answer. More complicated questions. Mary could use a drop down menu softer like sauce or SP _____. But the downside off this type off software is that they accept only off certain data for months. What if Mary would like to know at any moment what operations she's done on the data set, or to reproduce the same analysis on another data set that she have to start again from the beginning? These are examples Were American used a special data analysis program like our, where she can see at any moment on the steps perform on her data. She can easily change the data and even right custom functions. Toe answer the most specific questions. If we look at this light from left right, we can see that the most user friendly but the least capable tours for data analysis are spreadsheets, looking at it the other way around. Special data analysis programs are not the most user friendly tools, but they do offer the best options when it comes to predict these mothers are is a programming language, and the free software environment for statistical computing and graphics supported by the Our Foundation for Statistical Computing are. It's not like a general purpose programming language such as Java or C, because it was created by status additions. As an active environment, interactivity is a critical characteristic, allowing us to quickly explore our data. We can import data from different sources such as text files, appreciates databases or from you up, and then start creating graphs and performing

powerful analysis. Our allows us to start our working scripts. Scripts can be added and executed at any time, making our work reproducible. Another benefit off. Using our is a huge online community we can find online. Documentation and tutorials are is free and open source, which means that we can see the source code. It runs on all major platform, and it's easy for developers to share their software implementation off New data science techniques, for all its benefits, are has its share off shortcomings essentially based on S are is getting old, and if we would want to build it up, are probably one be our first choice. Another drawback. It's that the objects that we work with must be storing memory and working with huge data sets can quickly deplete our resources. Also, our runs only a single tread. The large amounts of data can overwhelm the interpreter. Some of these problems have been resolved by improvements in technology, but as data sets became larger in larger, they still remain a limitation. However, there are special methods to overcome these issues, so we don't have to worry about this for now.

Integrated Development Environment

[Autogenerated] are is both a programming language and the development environment for statistical and graphical techniques. It's statistical techniques include leaner and non linear modeling classifications and many more different types. Off floats are essential when doing exploratory data analysis, according to Wikipedia on Integrating Development Environment or I D. He's a software application that provides comprehensive facilities to computer programmers forced off their development. There are many options when we want to write code using our language, we can choose to work on our local machine or in the online environment. If we would like to work on our own computer, we can use our or our studio online. We can use Jupiter notebooks from Asia or our Studio Cloud. There are some other options as well, but these are the most widely used on our is quite a modeler programming language, the core component, which is required by every program returning our it's called based out. The score component contains only the core important beats, which may our code to run successfully. If we'd liketo add more functionality. We can extend this component by selling packages. These packages are collections off our functions data and compile code in a well defined for months. I mentioned earlier a couple off ideas, so let's start exploring the interface off each of them. The 1st 1 that will tackle it's our than our studio. And finally, for those who prefer nothing, style, any softer on their computer will take a quick look at our studio online and Jupiter notebooks. Let's start by first downloading our open your browser and goto chron dot are minus project artwork. Cran stands for comprehensive Archive Network. Here you can select the right version for you are it's available for Windows, Mac and Lennox. I will be using windows throughout the score, so I will pick the windows in stellar and

on the next page I will select. This is what you want to install are for the first time after this All the steps during installation are easy and straightforward, so I'm going to skip them once you ve styles are you should have a window open that looks similar to this. This window is the console where you can see information such as the version that you just don't want it in the output that results from running your code. We can write and execute code here, just as in a shell fashion. Or we can open a new script window. Why going to file and then clicking on the new script bottle. When we write in the council and he to return our expression, it's evaluated. For example, if we type two plus two, then the result should be four, and indeed it is. Scripts are very useful. They allow us to save our work and edited later, we can perform the exact same operation that we did in the console. If we type two plus toe and then he the control and our keys, the result should be printed on the console. The most common application for creating analysis with our it's actually our studio. Our studio uses the our version that we just installed toe down landed. We have to goto our studio dot com. Here. We have to select our studio down, run our studio for that stop. It's free as well, so we don't have to worry about spending any money fast forwarding the installation process. You should end up with something like this. The screen is divided into several sections forced on the top left. We have two designated space for our script. Once we created from file menu being it'd we have the console, the place where we can see our results or our errors. The next section shows all the objects or data sets saved in the memory. In the last pain, we can see our graphs. Our packages or the help window are has 1,000,000,000 help Any time when you don't know what the function does. Type question mark followed by the function name. To find out more about some function, we type question, mark some and run the comment in the bottom right. We get an explanation off what the some function does. Return the storm off all values. So some from 1 to 4 we'll perform the some off. One, 23 and four, Which is them. If you work on different computers and you want to access your skates anywhere, and any time you can opt for Jupiter notebooks from Asia or our Studio cloud for the 1st 1 you have to go to notebooks. The Asia that's come create a Microsoft account and then, under my projects, select a new notebook, give it a name and choose our as a programming language inside the notebook, we can have text commands and graphs to run a command. Just hit the wrong button. The second online tool. It's our studio online. We can access the story by typing _____ with your dark cloud on our browser. As you have probably noticed, we need to create an account in orderto access. This idea, our studio online looks exactly the same, like our studio that we just start on our computer.

Variables and Operators

[Autogenerated] are is a powerful scientific calculator. We can perform math operation using function like some or by using operators, as in any programming language. If we want to save our results for later, we'll have to create variables. Imagine that we go shopping and we want to keep track off the character value as we are or remove products. The most common option to do this is by creating a variable. The very word is a container that stars of value. When we enter the shop. Our basket, it's empty. After we add an upper, the cart value becomes, too. In our we use the Assignment operator, which is the arrow sign to stay toward the basket Value Ladies Basket Value Assignment operator to adding a cookie or pizza and a lemon increased the basket value to 20 according to Wikipedia. On assignment statement. Set or resets the Value store in the storage location, denoted by the variable name. As we saw, we assign a value toe a variable name using the arrow sign. Using the bring function, we can see that the products variable contains only the text Apple there Simon operator is a command, and it is telling the computer to assign the text apple to the product. Variable another way off. Doing these is by using the sine function. The first Parramatta is the variable name products follow by its value apples. By running the print function again, we get the same result as with us Simon operator. We can also use the Simon operator in another way by force, declaring the value and then declaring the variable or by using the equal operator this way off. Assigning values to very worst can be extremely confusing, so I would recommend not using it. Being a statistical programming language. Are can perform complex math operations. These operations are met up off smaller ones. Using arithmetic operators, we can perform basic math. The blast sign. It's called the addition operator. And just as its name says, it's some upto value. Any operator that requires two para mater's. It's called a binary operator. All the math rules and operation that we learn in school when we were young are present in our it's obstruction. It's represented by the minus sign. Multiplication and division take priority over addition and subsection with star as a multiplication operator and slash as a division operator Modelos and Interred your division are other useful operations. What about comparing tau? Values cannot do that off course. We have logical operators that run a logical tests and return either true or force. As a result, let's compare the prices off to fruits, banana, an apple, a banana cost $2 an apple 2.3. Which fruit? It's more expensive, most expensive. It's a very well that stores the output off apple greater than banana. Isn't April more expensive than a banana where it is so the results will be true. The less than operator works in the same way carrot and lemon prices will be compared. This time, it's a lemon less expensive than a care the answer through their ___ that it is to test. If a pineapple has the same price as a potato, we can use a double equal operator, which, in our case, it's room. On the other hand, if we'd like to test if they have different prices, we can use, the is not equal operator in our case, the result. It's false. In reality, there are many situations where we want to do. Multiple is best either carried more expensive than an apple and

more expensive than a pineapple in our health. Come end or and not operators all off name, take two values and return through or fours. The operator eater's true only if both barometers are true. So true and true. It's true, true and false. It's false force, and True is false and false and false. It's false. If at least one value is true, then the our operator will turn true. But when both parts are forced, the results will be falls. The not operator negates the value off the perimeter. Under true becomes farce and force becomes through. The first thing that we're going to do is to perform some simple math operations. With hashtag, we can write comments like my calculation. Let's start with addition to Plus four, and then we can substructure one, just like in met. If we want to prioritize this calculation, we need toe at Parent Is this before multiplying? Get by. Then, for example, in order to run this line off code, we click on the run bottle and the council prince. The results 50. We can save the output in a variable called results when saving the calculation in a variable are doesn't bring anything because it assumed that we will use this variable sometimes in the future by simply typing results. The consul prints its value. 50. When illustrating the mat operation, we use the sign operator instead. Off this, we could use the sine function. The first argument is the variable name between coats Result toe and the second argument is its value. Two plus four minus one. Everything multiply by 10. Let's test if results and results to are the same to do days will use the double equal operator hit, run and we get through as an answer. We can assign 50 or any other value to multiple valuables in the same time as follows Calculation or Simon Operator Result Assignment operator 50. The calculation and results variable will be equal, both storing the value 50. Beside the well known operators, we can use functions like Dorial or a B s. A B s function computes the absolute value and the factorial calculate the factorial off a given number. So a B s off minus four. It's four and factorial off four is 24. We can now move to the logical operators let me add a new comment in order to dealing it. This section from the arithmetic one. As mentioned earlier, these operators compare values and return a result off. True or false. We can try something like this. It's one equal to one double equal sign is a logical operator, and in this case it returns true to greater than five and seeks less then for both returns false. What about to greater than five and five greater than one? The square is returns force because to greater than five, it's false. But to greater than five or five greater than one will return. True, because at least one of the comparison returns true if we want to negate a variable we will use is different from operator. Is our results very well different from 55? By running this line off go, we knowthis that indeed, the result. Variable. It's different from 55

Data Types

[Autogenerated] later, it's the center off data analysis. Without data, there is no analysis. Every piece of data that we are working with has some characteristics. These characteristics can be summarized under data types we wantto analyze marries behavior during shopping. The first data type is character in some other programming languages. This type is known as a string. To create a variable off type character, we have to put our value in quotation works. Protection marks are used to distinguish between variable names and the actual values. Anything between coats in our will be treated as a character. It doesn't really matter what we put between them. For example, typing number one surrounded by coats makes it a character. We can use single or double coats to enter data, but not both. At the same time, data science is all about numbers. Number thing are are by. The four story is double working with fractions off. Whole number is a unique feature off double. This translates toe, allowing multiple digits after the decimal point, buying two apples at $2.3 with each results in a variable off type double, another type of numeric number, although less frequently use it's called Inter Jen. This type is actually a simplified version off double. Using this type, we can only store whole numbers without decimal components. If we want to save a number as an integer, we must are the capital letter l after it without the capital letter l. R. We store, the number is double. We can check the type of variable using the type of function, and we can even taste if a specific variable ease off numeric type or not by using the east dot numeric function. Now you're probably wondering what the difference is between quantity and quantity integer. The only difference is how the objects are starting memory. Integers are defined more precisely than doubles in your day to day job. You would want to use double instead off integers. We've talked about this counts. Is there any discount that can be applied to apples? The answer can be either true or force. The next data type can actually help us to work with these kinds of values, and they're called logical or Bolan. Let's assume that we bought the apple at full price, so the discount it's set to falls. There are two ways offsetting a logical variable as true or false. The first way is by assigning the false value directly to the variable. No codes are needed. The other option will be just the time. The letter T, which stands for True or the Letter F, which obviously stands for false. These data times are the most common and the most frequently used one beside them. In our we can also have complex numbers. An example off such a number. It's two plus e to represent the real part of the number and seeks represent the imaginary term role is another not so popular data type, and for good reason. It's not easy to create very birds off a road type. We actually need to use the row function to create one. As a result, off calling dysfunction, we get the row bites off datum. All the phenomenal data types are called Atomic Vector. Data Time Complex and RO are not widely used, so we won't cover them in this course. But it's good to know of their existence. Let me ask you a question. Can you tell me which data type issues by the product variable or the quantity is this count character type or a

logical one. You've probably figured out all the data types almost instant entry by yourself without the need to exercise at the type of the variable. This process is called type inference. In our we do not need to specify the data type because the interpreter will in for it. And by the way, those hashtag position after the variable are just, um, Sipan. Comments are will ignore every bit off text that is positioned after that until the end of the line. Mary, our character will help us to understand better. How are works? Wants to analyze her behavior during shopping. The first thing that comes into our minds when we talk about shopping are the products themselves. Well, we can create a product variable to store all this information. We are going to start by typing a comment where we specify what we're going to do. Create a character very one. Using the assignment operator, we assigned the value of cookie dough products by writing the variable name and pressing under. We get its value printed outing of counsel. Now let's ask our toe, tell us it's type using the type of function the argument. It's product or what we want to test. As we can see product have the type. Character is expected. Quantity and price are other useful information about products. We can check their type again with the type of function and just as expected, they are doubles. Remember what we said about doubles in the previous video. By default are saves number doubles even if their whole numbers. We can start nomadic values as doubles or integers, and the difference between them is how our store them in memory. Because we can have only holy numbers for quantity, we cannot buy two and 1/2 cookies. We can memorize this variable as an integer in the quantity integer variable by adding capital. L. The type of function points out that this variable is an integer idol. If we look at the output, we don't really see any difference between quantity and quantity. Integer. The is numeric function checks if the very birds are or not numerical and it returns a bowline answer. True or false are even contains a specific function that allows us to test it for very bodies. An integer or not, this function it's called East dot integer and just as its name says, it can determine if a variable is off type individual. Analyzing the results off type off is the numeric and is the integer. We noticed that integers are always numeric, but numeric are not always in. Voyager's didn't marry by the cookies on this count or not, let's assume that she bought them are full price. So with the sine function, we can set the discount. Oh, Falls Force doesn't have to be surrounded by coats, but it has to be all in caps on equivalent way off Writing. This is by using the shorthand version off False, which is just the capital letter F.

Code Style

[Autogenerated] you may be wondering what kind of rules we have in our and when I say rules. I mean suggestions on how to structure your coat properly to make it clean and possible to maintain all programming. Languages have their own guidelines on how to deal with valuables.

These guidelines make our code more readable When creating variables. Try to find meaningful names. If someone else tries to read your code, he or she should be able to understand it without asking you any questions. Setting a name as Apple where we would like actually to present products In general, it's not the best choice. We could name the variable products instead, I think 52 basket. It's another bad example. What? It's $55 5 products. It's not clear. Instead, we can rename it basket value, and we know that we are talking about dollars and not quantities. What about the letters in our its base To use only lower case letters using Paschal case or camera case might be better suited for other programming languages, but in our it's always preferable to use on Lee. Lower case letters are it's case sensitive, which means that this variables are different. Using only lower case letters makes it easier to create consistent object. Special characters like dollars are another Don't when creating variables or objects in our, they will make our code hard to read products. Dollar, Doral, Doral and Basket are silly examples, but you get the point. Keep it simple and clear. Last recommendation, but not least, avoid function names when creating variables. All writing some or mean function will create errors later, when we will actually want to calculate the sum over a few elements. Comments have the power to make it easier to share an understand cold that was written by someone else in Are we right important findings or decisions regarding the model chosen. But try not to explain what each function and each line off goat does. Toe actually make some sense off this rules. Let's go over a terrible example off writing on our function and try to make it more readable. What is the first impression? When you look at this code, it is clear what the function does. We have several, if statement, a division, a sound in dysfunction. But what is happening when the condition it's false? Certainly it takes some time to understand what's going on. Let's change the code into something more readable. First we saw that the recommended assignment operator. It's the arrow. We use spaces before and after it moving forward. We can aunt a function argument on the same line and regarding the if states months, we should place on Lee the condition in the same role. What about all these brackets? Well, the title of a style guide says that the open brace should be the last character on the line are so the closing brace should be the first character on the line, and the content should be intended by two spaces. Are doesn't need semi currents. To know that the command is finished, it will figure it out that by itself, the last step is to change the function name into something more meaningful, perhaps average if we compare the first version off the code with the one. After all the changes made, you can understand why indentation blank lines, bases and brackets are essential. These are known as white space

Organising Your Code

[Autogenerated] if there are good practices on how to organize our code within a script. There are also good practices on how to organize ourselves while working on multiple project or scripts. It is recommended toe have our scripts in the same folder with our data set. As we work on more and more projects, the number off scripts and data sets increases. How can we keep track off which data belongs to which scripts If we saved them in the same location, our studio provides us ways to organize and manage our script by allowing us to create project. For now, you can think off a project as a directory where we can save all the fires that belonged to one topic. We can, for instance, create a budget analysis project. This project includes our script, our data, set graphs or even reports adding or eat me text file will help the people with whom we share the project. To understand better the topic. We are not limited to one project, so we can set up another direct story for product analysis and another one for customer behavior analysis. Let's see how to do this in our studio here. I have the code that we using a previous clips in the top right. We notice that we don't have any project assigned to this script by clicking on the small arrow and selecting new project. We get this window, we can choose between a new directory or an existing one. I didn't create a specific folder for the script, so I'm going to select a new directory and the new project. Next, we have to name our project. Try to give it a meaningful name that expresses what you are going to do in this project. Then we have to select the location where we want to save it. Once you've created the project, it will appear in the upper right corner. Another fantastic future off project is that we can open multiple scripts at once with different data sets. And in the Environment section we will see only the details regarding those scripts.

Summary

[Autogenerated] we have talked about our is a programming language and as an environment for statistical computing and graphics are is used by programmers and data analysts and data scientists as well one off our strengths. It is amazing. Graphs in just a few lines also are has implemented a huge normal, off statistical algorithms. Next, we comfort different environments where we can run our code are in our studio, our local versions. But there are some online tools, like our Studio online or Asia notebooks, which facilitates writing code from anywhere in any time. The only requirement is to have an Internet connection. Then we perform some basic math to discover the arithmetic operators, such as addition or multiplication in the logical operators like, greater Than or less than working with data requires an understanding off the difference between data types in our this is even more important because we don't specify the type of variable. This process is known as type inference. All the data times are quite straightforward except the difference between doubles and integers. By the fourth, our store's numbers as doubles, even if

we work with whole numbers and the only difference between doubles and integers is the way that they're stored in computers. Memory. Last but not least, we discover howto organize our coat within a script and between different projects respecting right lines regarding variable names. Spacing and comments will facilitate your work, and you'll make your coat more readable when sharing it with other people in the next Ma June were discovered data structures available in our and start doing operations with factors the simple data structure.

Exploring Vectors and Factors

Introduction

[Autogenerated] hi and welcome back story later, using variables. It's useful when doing small task, but it isn't when we would like to work with big data sets. Vectors are the simply is data structure that can be used to store information. We're going to see what data structures are available in our and what makes them different. Then we will focus on creating vectors using different methods. Toe make vectors useful. We will order, sort and retrieve elements based on specific rules. Next, we're going to try out some operations that can be run on sets and discover why factors are useful when working with categorical data.

Data Structures

[Autogenerated] we use different, very burst the store information about Mary shopping behavior such as product or price. This is not the only way to collect data in our in our day to day job. We work with objects called data structures. These data structures, as their name suggests, represent a way to organize and store data, to facilitate different operations and to perform faster calculations. Marries basket. It's a collection off products. What matters? Do you see within this basket? How can we make it more structure? Well, first, we can group the products by their nature into two categories. Products of the same type and products off different times in our the collections. Off data that have the same type R vectors, which are the simplest data structure. Factors which are used for storing categorical data, a raise and mattresses, which are nothing more than a generalization off vectors, the specialized, structured or objects that store elements off different data times at least and date of rains least are one of the most complex objects

because they allow us to store even other lists. We can think off data friends at spreadsheets where our data it's organizing. Two columns and each column has its data type within a data frame. We can have all kinds off data types, but within one column we always have one type other crazy at the group, our product by our their dimensions, there are one dimensional objects, two dimensional objects and and dimensional objects. When I say in dimensional, I mean more than two dimensions. Victor and lists are one dimensional structures. Mattresses are pretty marginal. Structures use mostly statistics in a data frame. Each column or represents a variable, and each row represents an observation. All columns must have the same length. Cherries are the only objects that may have mourned in two dimensions. A Metrix is an array with two dimensions.

Creating Vectors

[Autogenerated] creating Rector's is not a new topic. We've seen it already, but you didn't know they were vectors. We created many very well so far, but in our they're considered factors. Electors are a collection off elements that have two properties. They have one dimension and contain one data type. We create vectors with the sea function and with a Simon operator, our first defector hiss five elements ranging from 1 to 5 to see its components we call the prin function. This matters looks very familiar, right? We use the same method to create a product or price variables. In the previous module, we generated variables with the same function, which means this is another method of creating vectors. The first argument. It's the director name, followed by its values, the coroner operator. It's a shortcut for creating vectors off consecutive numbers. The previous vectors, with elements ranging from 1 to 5, are constructed with less code using thes operator. This approach is convenient when we want to create a factor off consecutive elements ranging from 100 to 1000 generating sequences that respect the given pattern saves times as well the SEC function. That's exactly this by generating a vector off a settling following a specific increment. The first argument gives the start of the Vector one, while the second sets the end off the sequence. Five. So far, nothing has changed from periods methods. We have consecutive numbers ranging from 1 to 5. The next argument by represents the increment. The first element. It's one and the next one. It's one plus 1.5, which is 2.5. The last component off the vector is for because four plus 1.5 it's 5.5, which is bigger than our upper limit of five are got us cover. If we want to believe sequences off the same element, or to be more precise by replicating the same element as many times as we need rap function, close the values given as the first para _____. For as many times as the Times Argument says, here we repeat the numbers from 2 to 42 times. The scan function allows us to create vectors by introducing their values directly into the console online. The combined function. We don't insert the arguments inside the brackets. The

empty parentheses are important because without them, our research for 100 course can by person wrong were redirected to the console, where we type each element in a new line to finish this operation Weekly. The Enter key twice every data structure has its attributes. Each director has a length that represents the number off elements, any number, according by itself, in the expression, it's a factor off length. One. We can name it element of a factory. This is an optional attribute and names one change the vector values. What will help us to select different components, The data typed another mandatory at route the vector type it's given by its elements. There is a function to help identify each of these attributes length names, type of. There are six victor types. Character, double into your logic complex and roll complex in roll are not widely used, and we won't cover these types of vectors. Let's create a few vectors. Normal toe help Mary to store information Using our studio, we will check the type and the length of a vector while giving names to its values. Using the Sang command, we create a product vector were restored for products Apple, Cookie lemon and Big Sam. The East. The vector function confirms this data structure is a vector. By running the type of function we obtain the vector state a type character for character vectors. We can see the length of the vector, but also the length of each component. With Ngor. The product Vector has a length of four, and the first element has five characters. Let's start the price information in a novel factor. We have one price for each product, so the length of this vector will be equal with the length of the product Vector. Let me introduce you to a new function. It is called str and show the vector type, and if there are any additional attributes, it will show them as well. Four vectors. We have one optional attribute names speaking off names. There are multiple ways off assigning names to our prices. If we have 100 prices, how can we easily see which price belongs to each product? Not very easily Right? Setting the optional argument names is a good idea. When creating the vector. We specify the names as follows. Apple equal 1.3 cookie equal to and so on. Omitting the code works as well if we already have the vector created and we don't want to modify it, the name's function. It's a good choice. We assigned the vector product two names as an argument. Names takes the vector toe, which we want to give the names. Now A CR is more interesting because it shows the name's attributes as well. Using the scan function, we generate the quantity vector. Once we burst, run in the console, We've seen number one, followed by Collins. Here we input our first element toe the next element E six, followed by three and four To stop the function press enter twice.

Manipulating Vectors

[Autogenerated] Now that we have identified several metals to create factors, let's discover how to manipulate them. The manipulating process consists of several operations such a sorting,

ordering their components and selecting different elements choosing items from a fight or comes in handy when we work with the large data set. And we want to create a subset off the day down, starting and ordering my seem the same to you. But they're not the same. This is our original quantity of actor with five elements. The sore function changes the order off the items from the smallest, the biggest one, and we get the vector that consists off 2457 and eight. On the other hand, the order function returns. The indices were the smallest element is in the original vector. The smallest element is too, and it is the 3rd 1 in the initial vector forest higher than two, and it is the fourth position in the original vector. We can select animals by their position in our the first element has index one and not zero. Rules contains logical operators and returns only the items which satisfied them. Another medal, it's using the names that were attached to the vector elements. This is my thought. It's not working all the time because names are optional attribute for vectors and without setting the names. First we get an error. We have a veteran five elements to select elements by their position. We use square brackets inside the brackets we can have positive or negative in. This is the positive in. This is our used to specify which items to select. Having toe between brackets, we returned the second element from the left to select consecutive elements. We use the Koran operator as follows are from 4 to 5. Return the items at Positions four and five to choose various elements but not necessary. Consecutive will use the sea function. This nine off code returns the second and fourth elements. The negative index says the position Toby skipped in the whole selection, writing in between square rockets, minor stool will return or elements except the 2nd 1 We can omit wonderful items as well, by adding the minus sign before declaring the positions to be excluded. Declaring rules since I parentheses helps us to quickly discover values that are under equal or over a particular value. Here we use logical operators to see the value that it's equal to one. We write any double equal one or to see the elements that are greater than four. We use the greater thing operator to deal with multiple values. We need the matching operator and to specify the rule. If we ended names to our vector, we can retrieve victor elements as well. Here I created the names for our vector to select the vector components, using the names we type the names between codes like this, we can manipulate the vector that we created in the previous demo. Which product is the most expensive? With the Arab woman decreasing set to true, we are able to sort the prices in decreasing order. Pizza being the most expensive product bought by my room by default sort comment will arrange our elements into increasing order. Let's have another look at prices. 0.5 is the smaller surprise, and there's 1/3 component of the victor. Nine is the higher price and is the last element of the vector Order Command retrieves the order in the original vector, the first element history because smallest value 0.5 is the 3rd 1 In the initial vector, we can get the highest price with Mark's function as well to see the index and the product. This price belongs toa. We use the witch that Max Command

Peter the fourth Element. It's the most expensive one our provides mean and which don't mean function as well. Using the rank function we get the genesis for each material position happen is the second least expensive, so it will get a two. Cookie is the third least expensive product, and lemon is the least expensive. One tear for it gets a sign, a one with positive numbers between square brackets. We select elements from a vector product score brackets do return the second element from product victor cooking. The negative numbers between square brackets ignored the specified position. Swapping the first selection with a negative number. We will see in the console all the vector values except the second entry to select only the third and the fourth element. We use the sea function and to choose the lemon prices, we use the name between codes

Operating Vectors

[Autogenerated] We don't use Victor's only to store data and to order their elements. We can also do calculations you perform at operations between vectors and scholars and between two or more vectors. Remember Mary. She wants to know how much you would have to pay if she doubled her products. We have one vector to start, the quantity bought from each product and one victory prices to double the quantity we want. Apply the whole quantity vector by toe for the total basket value. We multiply the result with the price. All arithmetic operations are done element wise, this means that we'll end up with the vector that stores the total basket value with the same length as the quantity in price. The first element. It's two multiplied by one multiplied by 10 which is 20. The second element, it's 44 so on when we operated the vectors off the same length. The resulting vector has the same number off elements as the input in your life. We work with vectors off different lengths. The shorter vector. It's multiplied until it has the same number off elements as the longest factor. This operation is known as the recycling rule. What about a logical operations? They perform element wise as well, and the output contains elements as through and falls did marry by four or more product off the same type, comparing the quantity vector with four, we get a vector off. Five elements were the last two entries are true. We have to pay attention to the data type that the vector contains. What will happen if we put character and double in the same vector? We we get an error. We won't get an error because are we convert the daytime toe character. This conversion is known as coercion. Printing the output. We see the number surrounded by codes, which makes them a character. This implicit coercion is happening without warning, a vector that contains character and numeric elements. We be coerced into a character of actor when having a logical values and numbers will end up with a pneumatic vector where false zero and through its one and the vector with logical and character values will be

transformed into a character. One. There is another type of coercion, the explicit one. This one is executed using a specific function to change of actor into the desire data type of Iran The s the desire date that I've function and we are done to convert a numeric vector into character. We use the s, the character command. There are functions for each data type. Explicit coercion helps us to deal with incorrectly categorized data are in first the data type for a vector and one value possibly miss. Lead it into the wrong data type. Not all conversions are possible. We cannot transform American to character or character into a numeric. What will be the equivalent off? One into character doing illogical coercion Sze are we produce missing values? Missing values appear in our as any. Besides, next to our output, we get a warning message and nice, introduced by coercion. Let's put everything we have learned into practice here we have our full prices quantities and the new director that started discount for each product. To calculate the total basket value, we have to first apply the discount and then to multiply the result by the quantity our input vectors are length for so is the output. These prices are from store A and we can save them in another vector for the same products. We have the price information from another store starting price Tor be vector. Let's test if the prices are the same in both shops using number equal sign, we notice that only the prices for Apple are equal. The four prices factor is a double, but writing its elements between codes makes it a character. We don't have a discount value for the first and the third product instead of zero. We can write false using logical values with doubles in the same vector transformed force into zero. We can change the full prices maker back to America by running the s dot numeric function. Now let's change the second and third element off quantity into strings, trying to convert strength dramatic. We produce an A and we get this warning message.

Sets

[Autogenerated] set on a collection of objects that turned the same attributes. These objects are called elements or members and can be anything numbers, persons, colors and so on. Two sets that have exactly the same elements are called identical. We have total Merrick sets where the 1st 1 has elements ranging from 1 to 5, and the 2nd 1 has elements ranging from 4 to 7. Remember the common elements four and five. The process off farthing. The sets. It's called the Union, and it is completed by the union function, as are humans. We have our sets. Union returns on the elements from A and B. If they're elements presence in, both says the union will return the common values four and five. In our case, only ones, all of the elements after a union will be unique. Another operation is the intersection Intersection Prince. Only the common elements four and five. If we did addition with union, then we should be able to do differences. Were right, said Live. That's

exactly this. For Intersection and Union. We don't need to pay attention to the order off arguments. We can have a first and then me or the other way around and the results will be the same. This is not true for difference. The order was specified. The arguments in matters set diff between A and B means all the elements that are in a and not in big 12 entry at the unique elements for a on the other hand, this result, it's not the same with Seth diff between B and ES. The difference between B and ES will return all the elements that are part off being but not off a. Let's open our studio in Create two Sets A and B using the colon operator. The first set has items ranging from 1 to 20 and the 2nd 1 has components ranging from 8 to 30 for the Union Off A and B Returns or the elements, while Intersection Prince, only the common values set. This between A and B gives all the elements that are in a but not in B and said diff between B and ES gives all the components present in B, but not in a Let's combined set live between A and B with intersection and with set the between B and ES. Next, we can check if The result shows all the elements from a N B or the Union using double equal sign to get only once the value, true or false. We apply the old function indeed, combining the sets, we get only elements from a nd we can verify if our sets have the same elements or not with the set, equal function is the result. We get false because our sets have different elements to Turkey for elements from the first set, its present in the other, we use the matching operator. It returns on bullion value for each element were true means that the item it's present in both set.

Factors

[Autogenerated] factors are a special type of vectors. They're one dimensional data structure that store categorical information like product category. Its product belongs to a category, and there is a limited number of categories. We grew our products into two categories. One category is groceries and the other is electron ICS, transforming our images into vectors. We have two simple vectors products and category. So far, nothing new category. It's a simple character vector. To make it a factor, we simply apply the factor function when bringing the factor. We notice that our categories are not surrounded by coats as when printing a character vector and that we have a new activity. Levels levels represents the unique elements in alphabetical order. Now that we've seen what factors are, let's open our studio and discover even more to analyze. Wife actors are so special we are going to look at their structure. With str function, our encodes each level with an inter jer, and so all the competition are faster and needless memory Character takes more memory than integers electron ICS gets one and grocery gets toe due to their order or level. But what if we want grocery to be the first level. How can we change that when we create a new factor? We can specify the desire order like this. Now grocery receive one. An

electron is received. Two. If the level makes sense, we can even order them by printing the mascara factor that stores, milligrams, grams and kilograms. With the argument, order said. True, we get milligrams less than grams less than kilograms.

Summary

[Autogenerated] vectors are a one dimensional data structure that store only one type of data. There are different types of factors, such as character or logical. Besides, type vectors have lands and names as attributes. All the arithmetic and logical operations are done element wise. If we operate on Victor's off different length than the recycling rule will replicate the elements off the shorter vector until all objects have the same number off elements. Next, we had a look at conversions or the process of converting values into specific data types and set operations. Union Intersection and difference are the most common operations down on sets. Finally, we transform a vector into a factor and analyze its levels. In the next module, we will continue our adventure with our by exploring two dimensional data structure mattresses. Then we'll have a quick, big over a race, and finally we'll be focusing on one off the most complex types off objects lists

Using Matrices, Arrays, and Lists

Introduction

[Autogenerated] hi and welcome back. So far in this course we have look at different data time supported by our such as character or logical. And more recently we have tackled objects such as vectors and factors. Without a doubt, sometimes we'll need data structures more specialized in Victor's. This is exactly where matrices a race, at least coming to play. First, we will create and select elements from a Metrix. Then we look at how to work with and dimensional objects and how to create least the data structures that can hold even other lists. So without further ado, let's find out what a Metrix is.

Matrices

[Autogenerated] major cities are rectangular data structures with rows and columns, which makes them two dimensional. The most common type off metrics is America, but they can hold any data type as long that it is just one at a time. The first step in working with mattresses is to create one. We can create a Metrix using the mattress function or by combining vectors on roads or columns. Let's see how to create a magics from a vector. Here we have two vectors A and B that represent the product quantities bought from two stores. To obtain a Metrix, we can combine these factors by row, obtaining a Metrix with three columns and two rows. A nice alternative method. It's to combine them. My column. The output has two columns and three rows, each column being a vector. If the vectors Toby combined are not the same length than our will, repeat the shorter vectors. Tow the length off the longest vector in order to create a Metrix. This process is exactly like that off factors which you previously discussed. Analyzing our forced metrics. We can say that its length the number off elements. It's six. The six components out off type double and they are spread across three columns and to rose, which result in a new attribute. Dimensions to rose and three columns are so we can notice the roads have names A and B, another common operation done on objects. It's retrieving elements to perform this operation, we name our object quantity. To select an item, we have to specify both dimensions, rows and columns. The first normal represent rose, and the second number represents columns writing between brackets to comma three. We will sell like the element positions on the second row and third column nine listing only. The second number between brackets will bring All Rose and the third column. It works the other way as well. Only with two as argument. We will get only the second row, and all columns matrices are off the news in statistics and act similarly with Victor's, my performing Addition, substructure in multiplication and division element wise. Our input has two columns and three rows, and so does our output. The first element, it's calculated as two multiplied by a blast. Oh, for matrix multiplication. We have a special operator and the first metrics must have the same number off columns as the numbers off rose from the second metrics. Let's use vectors to build a Metrix. These vectors for quantities purchase from two stores A and being inside see function. We specify our vector separated by commas. The result. It's a Metrix with two columns and six rows where each column It's a vector. Notice that our metrics keeps the vector names on columns with urban command. We get an object with two rows and six columns. It Ro is of actor and has the vector names as throw headers. Using sit mind or air bind, we can add new columns or new rose to our existing metrics. Let's create another vector. See to store quantities from 1/3 store toe. Add it to the last metrics. We will first write a Metrix name, store quantity and then the vector Toby upended. The most common way of creating mattresses is by using the matter's function. Let's name it metrics quantity. We're restore quantities ranging from 1 to 18. Enero. Argument sets the desire number off Rose tree without

setting. By road to true, the metrics will be created on columns. By default. It's elements are read column wise as Victor's mother's is have length, names and type. Besides, these are magics has dimensions, rows and columns. Our metric store quantity has 18 elements off type, double disposed on three rows and six columns. Instead of using andro and and call functions, we have the short cut deem function for its structure. For my tresses, it is not practical to assign a name to each element. We assigned names or labels on Lee to the rows and columns adding names on columns. It's done by Cole Names function, and similarly, we have grown names to give names to rose. First, let's create a vector with Heather's for each column and then assigning tow the metrics. Now we have names for columns and rows. Our provides functions that perform calculations only on Rose or only on columns to such commands are row sums and cold sums. To find out how many products we purchase from each store, we're on the road. Some function we both 34 products from store A 28 from Store B and 23 from store seem to discover how many products we have from each type were on the Col. Sommes comment and see that we have 13 apples, 15 cookies and sword selecting items. It's done by specifying two elements separated by a common between square brackets. The first element, It's the roll index and the second element. It's the column index to choose the item on the fourth row. Second column We write one Comotto By omitting the column index, we select all columns to select only the first row. Well mean The column in Nice. It works similarly for columns writing on Lee. The second argument toe returns the second column. We can use negative integers inside brackets to omit a certain row off column. Here we select all elements except the second column. We can perform operations between a magics and the scanner or between two mattresses. Only operations are done element wise, such as an algebra. Let's bring the store quantity metrics one more time and then add toe. We are adding to tow each element. I think two mattresses will work as well. To multiply, two might resist. The first metrics must have the same number off columns as the number off rose from the second metrics. Sore quantity and metrics quantity don't respect this roar. So we will transpose the second Matrix using the T function here, Starr operator returns a different result than the Special metrics product operator. The star operator performs multiplication element by element, and the result Metrics has three rows and six columns. On the other hand, using the special matrix multiplication operator, we get a Metrix with three rows and three columns. Medics Multiplication keeps the Rhone names from the left metrics and the column names from the right metrics. Let's check if our mattresses are equal using the double equal operator each element, it's compared with each associate one here only the elements from the fourth row third column are equal In algebra. Identity metrics is very useful, creating an identity matrix in our it's very easy, giving the number of dimensions for as an argument to deac function, we get the magics with four rows and four columns.

Arrays

[Autogenerated] the last data structure that can only hold elements off. The same type is represented by a race. A race can have more than two dimensions. Actually, they can have any number of dimensions. Two dimensional arrays are actually mattresses, but a race with three or more dimensions don't have a specific name. Imagine that. We have information about quantities boat into shops A and B start as the magics. We repeat this process for 30 days and end up with 30 matrices that on columns have the stores and on Rose have registered the actual quantities, working with 30 mattresses. It's not an easy thing, so is that we could create an area. Our Day has three dimensions stores on columns, quantities on Rose and time as depth jumping on our studio. Let's recreate example from slides with only two days off shopping the quantities both on the first day, our store in the store quantity metrics created earlier. Let's assume that in the second day we doubled our quantities and this is our second metrics quantities Daito to create an array with the third dimension time we use the function array as input, we add. Our matrices followed by the desire number. Off dimensions. Three rows, six colors and two layers for time are prints each layer or each day in our case, separately from the first store a wee porches, seven apples in the first day. Wearing the A race attributes, we end up with details for length, type and dimensions. The shopping data array has 36 elements off type double as we already know, and three rows and six columns for each layer. We two layers in total selecting elements. It's another operation that can be run on a race to select all the products, both from all three stores on the first day, which are all the rows and all the columns from the first layer we write comma coma one If we want to choose all rose except the second row from the first column off layer to we write minus tool. Call my one coma to and the result. It's 14 and eight

Lists

[Autogenerated] the next doctor. It's least Lise are one of the most complex object times. They're one dimensional structures just like Victor's. There are several types of data structures in our vectors factors, matrices or raise least and data flames. We already talked about the 1st 4 So what are the least? Why are they considered to be one of the most complex objects? In our at least, it's nothing more than a container where its components can be any object. Even other lists we can have as many elements as we want least are known as records stiff structures because they're capable off soaring different types off objects. At the same time, we can create least using the least function where each element becomes a least element. Here we have two factors, each with two elements to select the first element off the least, which is the product vector. We use the double square brackets. Cookie is a component off the product factor, so it's a sub component

for the least to juice cookie. We have toe first pacify the vector position from the list. It is the first element, so we have one. And then we mentioned the position off the desire element cookie within the product vector. Now let's create a few more list in our studio here we have of actor and the magics from previous demo. We can create a least, bypassing them as the least function arguments least function. It's similar to the sea function, because each element becomes a least element. The first element is of actor and the 2nd 1 It's a matrix. As a result, we noticed that each least element has double square brackets. One between double square brackets Tell us that the first element off the least it's displayed and one between single square brackets shows the component off the first least. Element Seven is the first item on the victor store Quantity A. Where the vector is the first element of the list. We can refer to seven as a sub element off the list. Like the previous structures, at least can have names with names. We avoid confusions such as not remembering the kind of components we stored inside it were signed names with the names function after the releases generated or directly. When creating it like this, I created another least basket details where we have store name and product details as vectors. With so many different elements, it can be confusing to select only one component. We use the position off each element to subset them. We extract an element from the least. We use double square brackets. We can use either its position or its name. If it exists with names, we can use dollar sign to select elements as well. To choose the second element off the price vector we do this.

Summary

[Autogenerated] matrices are structures that have rows and columns and are two dimensional. While a race can have any number of dimensions, we explore the record See structures lists. These classifications is due to the fact that they can hold any types off objects even under least toe index A list. We can use a dollar sign or the double square bracket system. In the next module, we work with data frames and learn to apply several methods from the Thai reverse package looking forward to seeing you.

Working with Data Frames

Introduction

[Autogenerated] hi and welcome back off. All the data structures we have covered so far only least have been able to hold elements of different data times. In the same time, there is another

object called data frames that can do this, but with several restrictions. Before we discuss what this restrictions are, let's take a quick look at the module content. We'll create our first date a frame by respecting all the necessary restrictions. Then we'll analyze the structure by using special functions and by looking at the data and agreed for months. Next, we'll expand our knowledge of data frames by a pending new rows and columns to the original data, and by selecting different elements, finally really style a new package to increase the functionality off base are.

Discovering Data Frames

[Autogenerated] later frames are the most. Is it a structure in our A data frame is the least where all components have names and are of the same length. The easiest way to think of data frames is to visualize them at spreadsheets. The first row is represented by head arm, the _____ it is given by the least component names. Each column can store a distinct data type, which is called a variable, and it ro. It's an observation across multiple variables. Since they cut frames are like spreadsheets, we can insert the data how we would liketo There are many possibilities when inserting data. For example, we have product pricing, stor es and prices in starving. Is this a date? A frame? The answer is no, because the variable price it's not. In one column, it is divided between stor es and store beam. When you're arranging our data, the price is one variable and store is the third variable. Um, this is a date a frame with two variables and four observations. Data friends have attributes. They have the same properties as mattresses or other structures. Data frames have columns and rows and are two dimensional. All the columns have the same length and each column. It's a name vector. The vector name gives the data frame Heather. All objects in our have a class. A character actor has the class character, but for a data frame, the class is state suffering.

Demo: Creating Data Frames

[Autogenerated] let's open our studio and create our first date a frame. Then we look at it attributes and instructor, using special functions, one way of creating a data frame. It's by combining vectors. I have created several factors. Product, category, price, quantity, discount and discount _____. Now we will create a data frame called shopping data, using data that frame command. Inside this command, we add the above vector separated by Akamai's arguments. It is not necessary to create the vectors before generating the data frame. We can create items inside. It's such as the vector budget. Let me show you another way of looking at the data, then just the

simple print function. The function view offers a great view in a new window in our studio. We can search for a specific value in this search area, or we can figure the rose by selecting Flicker and then picking a rule. The column names are the vector names. They can be changed during the data for in creation by specifying the new name. Here will name the vehicle product. EXP the most common way of creating a data frame. It's by importing a file. I have the same data written in a spreadsheet to import its pressure. It we go to the file menu important data set and then choose from Excel in the new window. Browse for the file and click import in the console. We'll see the coat used to import this file. Now that we have created a data frame, let's understand this structure. The view. Could it produced by the Vue function was a good start. We could see the valuable names search and filled the rose when working with huge data sets. We look at our headers with the names. Comment. The first column. It's the product, and the last one is the budget. The _____ function returns the number off columns and the number off rose. We have seven columns with 12 observations by default, the head command prince, the 1st 6 throws off the data and the tell common priest. The last six throws off the datum. Both off them include the header and the index. In the beginning, off a TRO which represents the row number, we can actually select a specific number off rose to be display from the top and from the bottom off the datum by setting the an argument to the commands will return the first door and the last two rows. Another way to inspect our data. It's by running the SD at a function. The STR function will provide the normal off observations and variables. Very worst names such as product or category, the data type off each variable and the sneak preview off the data contained in each variable. Did you notice that our character vectors are safe as factors within the data frame? To change this, we need to send the argument strings. Inspectors toe falls and then we'll have our product and category columns. Memorize this character that's around the str function again to confirm this. Indeed. Now product and category are stored as a character. The type off in class functions reveal that our data said it's a least with a class date of ring. The summer recommends creates a small overview over each variable. For the nomadic variables, we see information such as minimal median, mean or maximum. This model's price $0.5 the largest price is $1000 with the average price being $260.5

Manipulating Data Frames

[Autogenerated] by manipulating data frames. We understand how to select elements are the new rows and columns and how to sort and rank Data frames are least where each element it's a name vector off the same length terra, for we can select elements in the same way as the least. We can use double square brackets or dollar sign data frames are also two dimensional objects,

like mattresses, which means we can index them as we do for my tresses by using the singles kind of brackets specifying the roll index, followed by the column index. You freeze upset with the single factor. The data friend behaves as a list and you for subset with two vectors, they be Havas mattresses. With these two properties, data frames can be indexed either as the least Aura's a Metrix using element position rules or names. First, let's remember how we stopped, at the least, using double square brackets or dollar sign to exemplify the elements election. We have a small beta frame called shopping datum to select the second column price we write to between dumber square brackets or the column name between codes. We get the same result with the dollar sign. Follow by the column name choosing the first and second items from product. It's done by using the colon operator inside square brackets. Only the third component from the Last column it's elected with either one off these comments. The second method of selecting items from a data frame is to use the single square brackets, as we did to self set metrics elements to select the whole product column. We mentioned only the index after comma that is related to the columns specifying only the name works as well. Writing. Only one is the first item between brackets. Prince the first throw the apple price from store, eh? It's obtained with one comma between brackets, or it won coma. The column name between brackets. We want to add a new column category to our existing data frame shopping date. Um, we can do this by using the same time function as we did for mattresses or by using the dollar sign there. We assigned the category of actor as a new variable in the data frame. The name off the new variable. It's specified after the dollar sign category by adding rose, we add new observation for a Data said, that contains product information. A new row it's determined by a new product toe upend the new observation to the data set. We run the air bind comment.

Demo: Manipulating Data Frames

[Autogenerated] Let's open our studio in practice elements election and adding new variables and new observation to our datum. First, let's have another look at our date. Offering data frames possess the characteristics off both least and matrices. We can subset our shopping data, which two factors and it will behave isometrics. Get the 1st 2 products we type under the road index. See from 123 and one under the column index to get only the third column, we mentioned only the second in extreme with one common tree. We print the price off the first product. The second group of commands is equivalent to the first time. However, there is a significant difference between singer square brackets and double square brackets by self setting. The date a frame with a metric syntax, we end up with the vector. Let's upset the first column with this method. Now to get the first column using a least upsetting syntax, we can choose between single square

brackets and double square brackets. Dumber square brackets produced a vector and single brackets produce a data frame. We can check this with east of Vector and is no data frame functions by using the dollar sign. We assigned a new variable store to our data offering these variable shows. If a product was purchased from Store A or from store, be to use the sea by method. We need toe first. Create the vector Toby a pendant. Here we have the price discounted factor, which is calculated based on the discount and price columns. Then we upend the price discounting vector to the shopping data. We'd see mine. We open new columns and with air bine were penned in euros. The new rose must contain all the details the other rose have, such as product or price. We already have two different data times, so we cannot create a factor and then added to our data frame as we did in the sea buying case toe at a neuro. We have to create a data frame where the name off each element is the same as our data frame. Heather's something like this. After creating the data frame with a single observation, Iran, the Air Bine Command with shopping data is the first argument and with the new product as the second argument

Working with Tidyverse

[Autogenerated] during data analysis, we spend most of the time on cleaning interest. Following the road data, we may create new vary bars, filter rose, perform summaries and so on. Daddy words. It's an unknown that let us to perform operations such as cleaning daytime, creating powerful graphs. When we started our we don't it. Only the base are to expand its functionalities. We need to insider packages. One off this package is is called Tai Livers. Diverse? It's a big package with several components will focus on the deploy our component, which concentrate on data manipulation. Usually we work with huge data sets and sports trends or correlation between variables. We focus our attention on the smaller later said. For this demonstration, we use a data frame with only four columns and six rows, and our goal is to reduce the number of variables even further. The Selectmen picks variable based on their name. The first argument is the data frame, and the second argument is represented by the column names to be selected. The outlook. It's a data frame with six rows and two columns, product and price. This key word or function select. It's known as a verb deploy our never changed the original data frame, which is shopping NATO. In our case, if we want to save the result of this query, we need to save it it under another data frame. By using their Simon operator, we can name this new data frame products selecting columns. It's useful, but sometimes selecting certain rose is even more helpful in our we have another world filter. The fetal function works similarly with select. The first argument is the data frame name product, and the 2nd 1 is the rule price greater than one. How many products have a price off less than $1? Only The third entry has a price for 50 cents. So after writing, this

command will have five. Rosen sent off six fingerprints only the roads that returns through tow. This calm person in the the row with the price of less than $1 was excluded. Inside the finger function, we can right all arithmetic operators discussing the previous module. It is also possible to combine multiple roars with a logical operators and all or not, we can you name this output data set to just in case we want to use it for further analysis. It is quite difficult to find meaningful names, right? What will happen if we do 10 or 100 selections in one day? We are not the only one struggling with this. Using the pipe operator, we can write multiple operation at once without renaming the intermediaries results. We can include the select and filter in just one query like this. We save our output under data said to name and from Sho Ping data. We first select product and price columns and then we feature the rose where the price is greater than $1. There are several differences between this line off coat and the previous ones. Obviously, we have the pipe operator at the end off each of our except the last one. We only managed on the data frame that we want to modify once and not inside off each verb the code. It's easier to read it because we focus on each function. We can really, at the set off actions from top to bottom, another toe function, that we're great together, our summary and grew. By using them, we can group our data into in smaller sets and then we can perform different operations like mean or count without grouping. The result will be just one line off cold by grouping hour later, my product will end up with three rows because we have treaties, things, products. Then for each item we calculate the average. Let's open our studio in Iran dysfunctions on our natives up.

Demo: Working with Tidyverse

[Autogenerated] we'll discover how to extend the base, are functional, kissed by installing a package, and then we'll use a few. Deploy our functions to manipulate our data. First things first. Let's in style die diverse. The command. It's very easy in Stein the packages and then input the package name, as mentioned earlier, Tiny raises a big package that contains smaller components. These components can be style on their own as well. We'll use only the component called D Player, and we can offer for installing only this one either option. After running, the comment will start the installation process. I previously tell them, so I'm not going to run these comments again. After installing, we have to load the package into our script with the Library command. Now we are ready to play with some, deploy our functions on the shopping date of ring. Let's elect several columns from our data frame and save the output into the product data frame. Now let's include only the products for which we had the budget greater than $200. Which products is the cheapest one? Arrange function sorts our data frame or sending by default to sort it to descending we just add the argument desk. Lemon is the cheapest products, and laptop is the

most expensive one. This price doesn't include the discounts on good time. The eyes to look at the final price to other knew very well. We use the mutate function inside this comment. We specify the name of the new column and how it is to be created pretty easy, right? All the new columns that we created are added at the end of our date. A frame Liz. The one more think, Let's calculate how many products we purchased in each category grew by and some worries coming to play for this task. We grew our rose by category, and we count the quantity purchased. We bought four products that belonged to electron ICS and 27 products that belongs to groceries, where the budget was greater than $200. But how many groceries were both from store, eh? Let's at the store is the second argument toe grew by function. There we go. We have 18 groceries from store A and nine from Star be

Data Structures Recap

[Autogenerated] data frames are the last data structures covered. So let's refresh our memory with data structures, classifications, we discover one dimensional, two dimensional and n dimensional objects. These data structures can hold items off the same data type or off different data types. Vectors are the simplest one dimensional structure that can store only one type of data at the time, such as character, double or logical. They're called atomic objects, since their components are all off the same types. The next one dimensional object is represented by least least are known as the reclusive structures, since they can start any type of elements, even other lists. Major cities are rectangular data structures, Withrow and columns Terra for their two dimensional. The most common type of metrics is the New American, but they can hold any data type as long. It is just one at the time. Data frames are the most use objects where each column It's a variable, and it rose an observation. The data frame. It's essentially a least where all elements are named vectors with the same length. Finally, or rays are the only ones that can have more than two dimensions. All objects that hold one data type are called homogeneous and all o. J. That whole multiple NATO times aren't called hetero genius.

Summary

[Autogenerated] in this module, we learn a lot. We created our first date, a frame starting with Victor's off the same length, and we selected elements using the least in the Matrix syntax. Then we start a package to extend the functionalities off base are we saw how deeply I r a component off tiny verse package works with a new operator pipe. We combined Deep Liar functions such a select Peter or grew by. There are many more functional than this, and all of them make more

sense on big deficits. Finally, we did a smaller recap off all the data structures, and we have two new terms homogeneous for objects that store only one data type and hetero genius for objects that store components off different data types at the same time. In the next module, we learn how to control the execution flow off our program. We will work with control statements such as if or else and with different types off loops such as four, or while looking forward to seeing you

Managing Control Statements

Introduction

[Autogenerated] hi and welcome back. The our interpreter process is called line by line. However, sometimes we want to process code in another order or two executed it multiple times, contract states months or Contra structures allows us to control the execution flow off our program. They are used inside functions or inside long expressions. We're going to see how, if else, and if l statesman's behaving are and what the difference are between them, then we'll explore an alternative to writing _____ else and else. If states, once known as the Switch comment finally will work with different types off loops such as four while and repeat lips.

Conditional Statements

[Autogenerated] the Our Language has available conditional statements that perform certain operation based on the logical result. True or false. The if statement is the most common statement and executes code on Lee. If the condition placed between bracket, it's true. Otherwise, the if statement will ignore that particular piece off. Gold toe Overcome the shore. Coming we at the optional statement else whenever the condition it's not mad are will run the code after the l skewered, Let's assume have a budget off $100 to shop online. The price and product information are stored into vegetables. If the price is less than 100 then we want our toe. Bring the message, adding productive card after declaring the very moment the next step is to compare the short price with our budget. 58. It's less than 100. So our prince the message that the short it's added to our car the next product cost $110 our condition. It's not met and we would like to add the product to our wish list to bring a message when the condition it's false. We at the l statement, this product, it's an it to our wish list. There is another form off. If else, which is the

vector rise form. This form returns off vector off the same length as the condition. Here we have a vector quantity with one and two s elements. The F S return. Yes, if the quantity it's equal to one and know if otherwise the result. It's a victory. Five elements is the quantity vector. Let's redo these examples in our studio with this statement. We get a message only when the price is less in the budget to bring the message over the Margit. Let's at the L statement now we're still missing one possibility. If the price it's equal to the budget, we get the wrong message. Product. It's over budget with others. If we improve our if state month and now we bring the correct message, the price, it's equal to the budget. The Asif statement must be written after the brackets. Otherwise, we'll get an error. We tested both conditions with Victor's off length one. But what will happen if the price? It's a vector off three elements. By running this piece of code, we only have one message printed price under the budget, even if the second and the third vector elements are greater than our budget. Next to our output, we have a warning message. If the condition has the length greater than one, then on Lee, the first element will be yours. This means that if it's looking for a single answer as true or false, it's Jack's the first element, and then it stops. We can overcome this issue by using either the all or any functions. If a least one price respect the condition, then the any function will return through. And the message at least one prices under the budget it's printed, on the other hand, for the all function to return through, all the prices must satisfy the condition the operators and and all I wisely used to combine conditions the single end or single or are applied element wise to vectors, while the double ent and double or are used for vectors are length one. By mixing conditions, we can check if the price is between two numbers. 58 is between 50 and 100. I saw it as a non victory ized version off or sees the first true value. It returns true, and as soon as it sees the first false value, the non vector rise and return falls. Let applied the verte rise version off the F I statement with If else we compare each element from the price victor to our budget by running this line off coat, we get five messages, one for each victor element.

Switch Statement

[Autogenerated] we saw that we can add as many else and else if statements as we want. However, having more than four is difficult to keep track of what is happening when the condition is true, on alternative to writing nested else and else if statements is to use the switch comment. The Sweet Command works with cases it seemed. Talks contain the value to be tested, followed by the possible cases. Here we have a function that converts the short version off the weekday toe. It's fully name. For example. Munn becomes Monday. The day function takes a value in and returns the matching case inside the streets command. There are five cases inside this comment.

If we passed you to the day's function, we get Tuesday, this works the same way for all week days. If we passed Saturday or Sunday, then the result is weakened. Switch can take either character arguments such as thes cases or in America argument by inserting five as an argument. We get the fifth case. If the number as an argument is bigger than the possible cases. No, it's returned. Let's do another exercise in our studio here. We have a vector and a function. The quantity of actor has four elements, and our function calculated the average By taking two arguments, quantity and type. We can calculate the average as arithmetic or geometry to help both cases included. In one function, we use the switch. Comment. Our Sweet has two cases. First, if the type it's arithmetic, then it calculates the arithmetic mean otherwise. It calculates the job. My trick mean by setting the type argument. Oh, arithmetic. We get 3.25 for the quantity mean and 2.78 with the Joma treatment.

Loops

[Autogenerated] you mention we have a regular with 1000 elements, and we'd like to print or to perform the same action for each element. Writing the same scene tax by hand 1000 times is time consuming and not reproducible to deal with these types off. Repetitive tasks are provide loops, but what are the _____? A loop. It's a sequence off instructions that is repeated until a certain condition. It's rich. The most common rope. It's the for Luke. Four loops perform the same task on all elements from the input it syntax is simple to declare the loop we start with the cured for then between parentis is we have three arguments. The first argument is the variable, which can take any name. Then we have the cure in and the last argument. It's a sequence or a vector off any kind. We read the slope like this for each variable in this sequence. Do this expression. Let's see an example. We have a vector cart with two products, and we'd like to read each element using a loop. The four loop looks like this for each product. In the cars we print the product our vector history products. So the low Peron three times the loop starts in the first row by assigning toe product, the first element of the vector cart apple. Then it executes the task. Printing the product. We have two products left so the loop continuous by going to the next product cookie and performing the same task it did with Apple printing. After printing the lemon, which is the last element in the sequence, the four loop stops, we can incorporate rules inside four loops as well. For example, we can print all the products that are not cookie again. The loop begins with the first item in the cart vector apple, then the test product. Different and cookies perform. The test result is true, so the apple it's printed next product takes the value cookie. It fails the test cookie different and cookie, and it's not printed. The four loop stops only when the all elements in the factor have been tested. Lemon is the last item in the cart of actor, and after

passing the test, it is printed on the loop stops. We have to be careful with four loops because in arm they don't bring the output in the previous example, we had an output because we explicitly call the print function without a print function for Luke doesn't save its output. Another type of flow pits the wild hope the while you performs the same operations as long as the condition is Strom. There, syntax is similar to Eve, my taking a condition and running a piece off court. As long as that condition, it's Matt while reruns the condition at the beginning off each loop. And if it's true, rerun the expression. Otherwise it stops to make the look stop. There must be a relation between the condition and the expression. Otherwise the loop will never stop, and it becomes an infinite loop. Here we have a variable called index with a value off one using the Y Lo. Let's bring the index value as long as the index is smaller than tree. How is this in? That's going to change if we declare it it as being one. We must increments our index every year in the first loop, the in Texas one, which is smaller than three and it's printed. Then the index value exchanged Toto and the loop starts again with the index set toe to instead off 12 is less than three, so the value it's printed and the indexes incriminated to train. Now the loop stops because the condition three less than three It's false. The last category off lobes is no repeat lips. They repeat the same operations until their start by hitting the stocky or by inserting a special function to stop them. The repeated loops are useful in angering optimization or in maximization problems as the syntax. They taken expression and then repeat the same action. Over and over here, we sent a value one toe X, and then we repeat the action off incriminating X by one until access pleasantry, the loop only runs two times, and then it is forced to stop by the break statement. The brakes statement can be used to terminate any York, and it is the only way to terminate a repeat _____. The statement is not the only statement that can alter the behavior off a loop. The next statement. It's used to discontinue one particular it oration and skip to the next iteration. We will see where these two statesmen are useful in the next demo

Demo: Loops

[Autogenerated] it's time to put everything we know into practice. We'll create different types off loops in our studio, and then we'll modify their flow by using the brake and next statements. Let's apply this count all off our prices. With a four loop, we can repeat this operation for all off the vector elements for each value from the price victor. We apply a 10% discount, and then we substructure the discount value from the original price in order to obtain the final price. Let's bring these lines off gold. There is no output. What happened? Four lumps in our bring their results on Lee. If we explicitly tell our to do this, otherwise it doesn't print any value to return a result, we have to save it in object during the loop. My saving the result in price discounting,

object and printing it. We get all the prices after the discount has been applied. Let's assume that we want to stop the four loop as soon as we reach a number that is greater than 100. Break statements do exactly this. The fourth loop calculates a discounted price for the first element 30 and then we start again, with the second element from the Vector 120 it's greater than 100 so the photo look will stop and the prince function returns only one price. Let's allow the look to move forward. Different element is greater than 100 by stopping the break function with the next statement. The loophole reads all elements, but only calculates and prince the prices that are less than 100 in the original vector. With this, the obstruct er, we don't have access to the prices in this is our A science and index to each element starting at one. So the first price has indexed won. The second price has in next to and so on. Usually, we note the variable with a small letter I, and the sequence is changed with the Kahlan operator to read the whole price vector. We write one colon length off price. Next, we need to change the code inside the brackets as well, because instead of value, we have I. So for each element, we applied discount and then we bring them. Let's create an index using the Y lo. We started by defining the index variable one. The next will have the same length as the price factor, and there's an output. We simply return the index inside the wild loop will use a Prince statement just to get a peek over the current value that is passing through the loop. Just as in four loops, we can use a break statement to stop declaration in here. We stopped the loop. Once the index reaches the value five. The last type off loops is represented by repeat. The Loves created using the repeat statements are infinite, meaning that they will never stop in this scenario, were actually forced to use a break statement. Otherwise, are we execute this program until the computer runs out off memory. Here we create another index ranging from 1 to 10. Once the index sistan the IV condition returns through and the loop stops.

Summary

[Autogenerated] we explore several methods for executing our code in an older other than line after line. Even L statements are the most common statements when we test element objects, while, if I statements are used for their victories, property, if else returns an object as a result, with the same length of the condition in our the main loops are for. And while the four low performs a set off instructions for each element off the sequence and the why Loop executes a set off instructions as long as the condition is room why loops can be easily transform into an infinite loops. Infinite loops can be forcefully stopped by using the brakes. Statement Next, and rake statements are useful when we want to skip or to completely stop the loop. Sometimes loops can be abandoned in favour off factories ations due to the fact that loops in our can be slow in comparison with other languages, especially the _____ troops which are loops inside other lips in

the next module will combine all the knowledge we have gained so far in orderto build. Our first function

Building Your First Function

Introduction

[Autogenerated] hi and welcome back. During this course, we have actually already use functions. We use functions to create vectors and matrices. Dysfunctions are written in our and have a similar structure to functions written by the user. Functions have a clear goal which is to avoid performing repetitively test. Adding these level off automation also allows us to eliminate user errors. The coat becomes easier to following adjust, which comes in handy if we need to improve it. Before we write our first function, we need to understand what functions are. We will then take a look at function components. How to pass data is an input to a function, and how to call them finally will create functions in our studio. Let's dive in and discover more about functions.

Discovering Functions

[Autogenerated] We've used functions many times throughout this course. But what exactly is the function? I want you to imagine the process off cooking a pizza in order to make a pizza. We need ingredients such as flour, eggs and salt. We need follow a recipe or rather, a process. We mix all the ingredients together and put the pizza into the open. After the process is complete, we are able to enjoy a delicious pizza. But wait, why are we talking about pizza instead of functions? The process to make a pizza? It's like a function. We place our imports the uncooked pizza. Inside a function, the Owen and we expect an output are ready to eat pizza. We can use dysfunction into two smaller functions that can be used for other tasks. The overall function can be reused for other types of food. Having destructor in place is what we refer to as a nested functions. A function is the sequence Off program instructions will perform a specific task. They allow us the ultimate repetitive tasks by making our code more readable. In our functions are objects and to create one, we use the function key word and the Simon operator. First, we need to choose a name cooking. Then, after the function cured, we write our arguments. Without a name, the function becomes anonymous. Anonymous functions are most frequently used as arguments toe other functions, the cooking function, things, the ingredients or the imports. As arguments, a

function can have any number off arguments. The arguments can be different or the same type between curly brackets. We write our instructions to execute our instructions. We time the name off the function followed by our inputs.

Function Components

[Autogenerated] you've probably noticed now that we have already used functions in the previous modules are comes with a set of pre created functions that we can use. Functions can be manipulated just like any object. They have several basic components. The first component is represented by the function arguments or as we saw earlier by their inputs. Let's have a look at the code of an existing function. If we only have the function name without parentheses, we can see the function code. Here we have the open function, which adds elements to a vector. The arguments are placed immediately after the cured function, and we can have as many arguments as we need by taking a closer look to the open function arguments. We noticed they're separated by commas, and the after argument already has a value. X and values are required arguments because we down them. The function doesn't work. We have a vector price with three elements, and we want to upend the value. Ethan. My writing price. Comitan. Inside open function, we get a vector consisting off four elements with the value 10 added at the end, but where our X and values our matches the input that we gave Price and 10 by position off the function arguments. In this way the price is assigned to acts becomes vote off them are the first argument inside the function. There is another Met off off, matching our moments in our by name, we can write inside open function, X equal prize and values equal 10. It is possible to create a mix off this as well. All three calls returned the same result. The apparent function takes three arguments, but we can call this function with two arguments. How is this possible? Another type of argument is the default argument here. The after argument already has a value. The length off X. This means that if we don't specify it when calling the function, the implicit value will be used. The default value for our argument is the most common value when we opened the new elemental vector. Most of the time, we want the new element to be placed at the end of the vector by typing the default value. Inside the function, we get the same result as without it. We are not bound to these different value, and we can change it anytime toe undervalue 10. After the second element of the vector, we stop the default value with toe. A function can have a special argument that dot dot argument happened doesn't have this argument. They don't know the arguments are used when would like to pass a variable number off arguments toe that function, for example, some and mean functions contained the special argument. The second component of a function is the body. The body is placed between Carly bracket and the represents the set off instructions. The function will

perform any function in our we return, the last result from the last line off gold. This can be changed by inspecting the return command any wearing the function. The last component is the environment. The function environment is the active environment when the function was created. Usually when we create a function, we do it in our work space, which is called the global environment. Why is this important? Are finds a value associated with the name using the environment. For example, we have a Faison that takes only one argument. X, which performs the sound off X and y. Why is not defined? Decided function. So our searches for the Y value in the environment where the function was defined. The global environment. There is a lot off interesting information about environment, but we won't cover it in these scores.

Demo: Writing Functions

[Autogenerated] We talked a lot about function components, So let's dive in and create our own functions. We're going to start by writing a function that adds to numbers. We call it odd because you want to add two numbers. Will have two arguments X and Y. The arguments can have any name as long as they're consistent in the body function. There needs to be a connection between function arguments and the task performed inside the body. After running these lines off court, we've seen it in the global environment. We have our function the plus size from our consul. Simply mean that our continues reading the code from the next line. And they're not part of the code itself. Calling the function with or without arguments, names gives us the same output. Three. If we said only one element inside the ad function, we get an error because both X and why are required arguments. Let's make wire default argument by assigning the value zero to it. Now, running the function with one element works. We can save the output express why in a variable called Z and perform on expenses return like this when we run the function are loads or the arguments, imports and only objects created within the function in temporary memory was the functional runs. We cannot accept ze variable or why value. Now let's create a function that returns the maximum value from a vector. The vector prices has 12 elements, and the new function it's called Get Marks is an argument to the function command. We passed X. How can we find the maximum value from a factor? We have to take each value and do a comparison. This is telling us that we need a four loop. We'll run the loop as many times as we need, and then we want to print the maximum value. This means the print function. It's outside off the four loop or return marks. We want to return the maximum value, so we have to define this variable. First, we define this variable before the loop and get the first value off the factor. Next in the loop, we compare our current max value to the second element of the vector. You're the second element is greater than the 1st 1 Then the marks variable takes its value. We do this comparison till we reach

the end of the vector. So inside the Fort Loop, we have I in two till the end of the vector. We can start in second position because we said the first element to the max variable already to compare tau values and the same time we need the if statement in the common vector element is greater than marks. Then this element becomes the new marks calling dysfunction on the price director, we get 1200 as the maximum element inside the price factor, which is correct. Let's bring the warning message if the factor prices contains negative elements, awarding message doesn't stop the function from running the task. If any elements are negative, then we print this warning message. Let's change the price factor to three elements and then run the function by calling the Get Max function over this factor. We get the correct maximum value alongside the warning message. There will be cases when by mistake, all the prices are negative and we don't want to run the loop anymore. Instead of the warning function, we can use the stuff function like this After running the function we get, only the error object has only negative numbers

Summary

[Autogenerated] writing our own functions. It's one of the strength in our because we can automate the repetitive tasks and avoid mistakes that arrives from coping and basting. Information the coat. It's also easier to read implementing. We define functions as sequences, off program instructions. That performer specific dusk. They're like any other object in R. We then looked at their major components, arguments, body and environment, without specifying the required arguments. The function return in error. This is not a case for the default arguments by omitting the default argument, the function we take. It's pretty fine value function bodies represented by the instructions themselves, and it is written between curly brackets. A function environment is the environment that was active when the function was created. In most cases, the world space. It's our environment and it is called global Environment. We finished the module by writing a function that returns the maximum value from a factor when has dysfunction by adding awarding message and then we even under the possibility to stop the execution off the function, the best way to improve your our knowledge is to practice writing code and analyze data doings. More cast ideas will consistently payoff. There is a big online community where you can ask questions and learn more about our. I recommend it to have a look at the cheat sheets provided by our studio to get involved into online projects and competitions and to participate to our conferences. This is a great way to meet new people that work with our. You can find data sets already in our or online a good place to look for data sets or to inspect notebooks created by other people. It's kagle for conferences. Visit our minus project of Orcs, last conferences or our

studio block. We can get in contact on Lincoln or on Twitter, and I'm looking for for your questions.

Course author

Mihaela Danci

As a data analyst enthusiast, Mihaela has special interests in translating vision into action using end-to-end data analysis. She is very passionate about teaching and resolving real-world...

Course info

Level	Beginner
Rating	★★★★★
My rating	★★★★★
Duration	2h 2m
Released	28 Feb 2020

Share course