COURSE 7- DATA ANALYSIS WITH R PROGRAMMING

R is a programming language that can help you in your data analysis process. In this part of the course, you’ll learn about R and RStudio, the environment you’ll use to work in R. You’ll explore the benefits of using R and RStudio as well as the components of RStudio that will help you get started.

### Learning Objectives

* Compare and contrast the R programming environment and the RStudio programming environment
* Describe the RStudio programming environment including its components and benefits
* Describe the R programming language and its programming environment
* Describe programming languages and appropriate use including examples
* Download and install R assets to a computer
* Open R and execute a command
* Differentiate between the R Console and R programming environments
* Execute operations in R using mathematical operators such as +, -, \*, and /
* Download and use RStudio Desktop
* Demonstrate how to complete basic tasks in R

THE EXCITING WORLD OF PROGRAMMING

[Introduction to the exciting world of programming](https://www.coursera.org/learn/data-analysis-r/lecture/NC1Mv/introduction-to-the-exciting-world-of-programming)

Hey there, data pro. You've come a long way since the beginning of your learning journey. Congratulations on your accomplishment.

Just think of all the skills you've learned along the way.

You now know how to use structured thinking to define a problem and ask the right questions;

**work with spreadsheets, databases, and tools like SQL to organize and transform data;**

**clean your data to make sure it has integrity before you analyze it;**

**create impactful visuals to illustrate key points;**

**and craft a compelling story to communicate insights to stakeholders.**

That's an impressive list of skills, but we're not done yet. Your skills set's about to get even bigger. In this course, you'll learn about a **new concept called programming and how you can use the R programming language to analyze your data.**

By now, you know that the data analysis process includes six phases: ask, prepare, process, analyze, share, and act.

Now, we'll learn all about the R programming language and how it can help you in each phase of the process. When you're done, you'll be presented with an optional case study. The case study will give you the chance to solve a data analysis problem using all the skills you've learned in this program. You'll find out more about this project later on.

Let's talk about computer programming. **Computer programming** refers to giving instructions to a computer to perform an action or set of actions. You can use different programming languages to write these instructions. You might choose a specific language based on the project you want to pursue or the problem you want to solve.

**The R programming language is super useful for organizing, cleaning, and analyzing data**.

If this is your first experience with computer programming, welcome. When I first started learning about data analysis, I didn't have a background in programming either. In fact, before I fell in love with data, I was trained as an opera singer. I also have a lot of friends that came into this field from the arts and learned about programming later in their career. **R is a great place to start**.

Learning R for the first time can be challenging and even more empowering. A lot of the skills you've learned in this program will help you learn basic programming concepts. Take it one step at a time and go at your own pace. Just like in previous videos, you'll start with the basics and move forward from there. You've tackled tough challenges before and you always come out on top. You've got this. Let me introduce myself.

My name's Carrie. I work as a research manager at Google. I lead a team that researches the best way to improve the performance of people in organizations. In other words, I help people work better and work smarter and help organizations function in a healthy and productive way.

I first learned R as a junior data analyst, while I was working on a multi-year project about virtual work. We were looking at data on people's virtual work experiences and trying to understand how working remotely impacts performance. It was a complex project with a lot of data to sift through. I kept encountering problems and searching for better and faster ways to do things. This is when I became aware of the power of R.

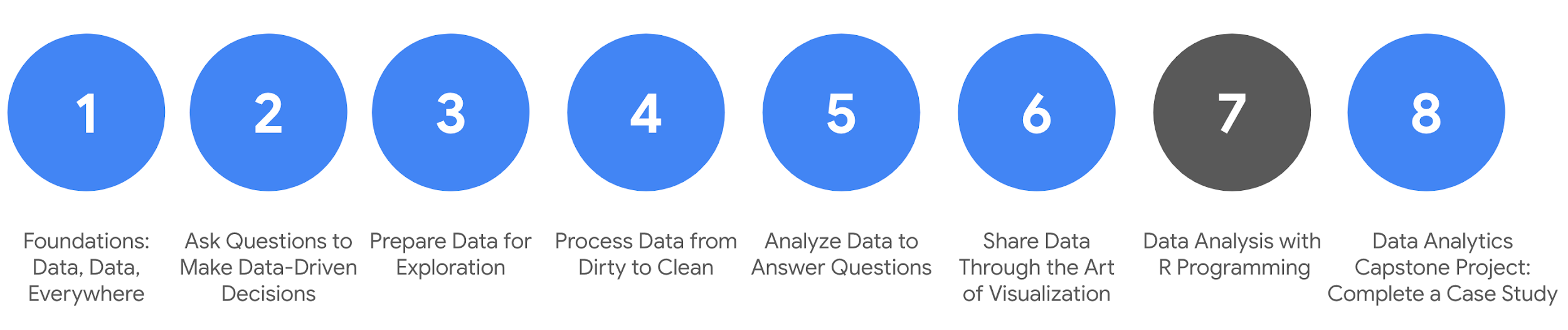
**Whenever I got stuck, I'd learn a little more about R and discover a solution to my problem.** I soon realized that R could help me do almost anything involving data even better and faster than I thought possible.

Fortunately, there are tons of great online resources for R and a super supportive online community. If I had a question, I'd go online and find the answer. As the project progressed, I was able to learn more and more and become a much more effective data analyst. My teammates even started coming to me for advice about R. Realizing that I could continue to learn my skills at any stage in my career was an empowering experience. Learning R unlocked my ability to perform data analysis at the highest level. In your future career as a data analyst, you'll have the opportunity to continually learn and grow. To me, that might be one of the coolest aspects of the job. Learning R is one of the most rewarding parts of that growth process. I'm still learning new ways to use R all the time. Plus, you can apply these skills to other programming languages like Python, Julia, or JavaScript. There's no limit to how far you can go with programming. It even goes beyond data analysis. After I learned R, I found myself thinking about all kinds of projects I could use programming for, both at work and for fun. It opens up a whole new world of possibilities. Now, let's talk about what you're about to learn. We'll start off with an introduction to programming languages. Then we'll take a closer look at R itself and explore its main features and functions. We'll also cover some basic programming concepts and learn how to use them effectively in R.

Next, we'll learn how to work with data in R. You'll discover how R can supercharge your data analysis skills and let you clean, transform, visualize, and report data in new and more powerful ways. Learning R will help you take your data analysis to the next level. It'll also look great on your resume. R is widely recognized as a key credential in entry- level job positions. Knowing how to use R will give you a big boost in your job hunt and will help you stand out as a new analyst. Coming up, we'll talk more about programming languages in general and how they can help you analyze your data. After that, we'll jump right into R. Before you know it, you'll be using R to power your data analysis.

[Course syllabus](https://www.coursera.org/learn/data-analysis-r/supplement/Y0Vr4/course-syllabus)

In previous courses, you learned how to use structured thinking to solve business problems; prepare, clean, transform, and analyze data in spreadsheets and databases; and tell effective data stories. As part of your growing skillset, you also learned how to create dynamic and interactive data visualizations in Tableau. Up until now, the skills you learned were closely tied to the features and capabilities available in spreadsheets, SQL databases, and Tableau. But what if you want to work with your data in more custom ways? Or what if the standard tools don’t have the exact functionality you need? This is when the R programming language can be super helpful. Using R, you will gain additional flexibility and control over your data and analysis.



1. [Foundations: Data, Data, Everywhere](https://www.coursera.org/learn/foundations-data/home/welcome)
2. [Ask Questions to Make Data-Driven Decisions](https://www.coursera.org/learn/ask-questions-make-decisions/home/welcome)
3. [Prepare Data for Exploration](https://www.coursera.org/learn/data-preparation/home/welcome)
4. [Process Data from Dirty to Clean](https://www.coursera.org/learn/process-data/home/welcome)
5. [Analyze Data to Answer Questions](https://www.coursera.org/learn/analyze-data/home/welcome)
6. [Share Data through the Art of Visualization](https://www.coursera.org/learn/visualize-data/home/welcome)
7. **Data Analysis with R Programming** *(this course)*
8. [Google Data Analytics Capstone: Complete a Case Study](https://www.coursera.org/learn/google-data-analytics-capstone/home/welcome)

In this course, you will learn how to use the R programming language to work with your data without tool limitations. You will get plenty of practice using R for statistical analysis, and RStudio—an integrated developer environment (IDE) for R that you will use to create advanced data visualizations with lots of detail. R makes it easier to present your data with beautiful, artistic style. A few other advantages of R include its:

* **Popularity**: R is frequently used for data analysis
* **Tools:** R has a convenient library of ready-to-use tools for data cleaning and analysis
* **Focus**: R was created with statistics in mind; data analysts can conveniently use a rich library of statistical routines
* **Adaptability**: R adapts well for use in both machine learning and data analysis projects
* **Availability**: R is an open source programming language

After you get comfortable and more confident using R and RStudio, you might find that you are curious to learn and add even more programming languages to your skillset (and resume). Pretty exciting, right?

## **Course content**

Course 7 – Data Analysis with R Programming

1. **Understanding the basics of R:** R is a programming language that can be used to perform tasks in every phase of the data analysis process. In this part of the course, you will learn about R and RStudio, an integrated developer environment (IDE) for R. You will explore the benefits of using RStudio to work with R. RStudio enables you to easily leverage the features and functionality of R.
2. **Programming using RStudio:** In this part of the course, you will explore the fundamental concepts associated with R. You will learn about functions and variables that you can use in your calculations and other programming. You will also learn about R packages, which are collections of R functions, code, and sample data that you can use in RStudio.
3. **Working with data in R:** The R programming language was designed to work with data at all stages of the data analysis process. In this part of the course, you will examine how R can help you structure, organize, and clean your data through functions and other processes. You will learn about data frames and how to work with them in R. You will also revisit the concept of data bias and how you can use R to address it.
4. **Visualizations, aesthetics, and annotations:** R is a great tool for creating detailed visualizations. In this part of the course, you will learn how to use R to generate and troubleshoot visualizations. You will also explore the features of R and RStudio that can help you improve the aesthetics of your visualizations. You will learn how to annotate visualizations and save the changes.
5. **Documentation and reports:** R has a number of different options to explore when you are ready to save and present your analysis. In this part of the course, you will explore R Markdown, a file format for making dynamic documents with R. You will learn how to format and export R Markdown and incorporate R code chunks in your documents.
6. **Course challenge:** At the end of the course you will apply everything you have learned in the Course Challenge. The Course Challenge will ask you questions about the key skills you have been practicing and will give you an opportunity to demonstrate those skills in three scenarios.

## **Are you already familiar with R programming?**

If you have used R and RStudio before, you might find the first two modules of this course a review of basic topics that you already understand. Feel free to skip these foundational videos and readings and proceed to the module challenges for Module 1 and Module 2. The module challenges will help prepare you for the course challenge at the end of this course. To earn the certificate, you need to score 80% or higher on all graded activities in the program.

## **What to expect**

You can expect to finish this course in about four to five weeks. That involves completing all the activities, including:

* **Videos** of instructors teaching new concepts and demonstrating the use of tools
* **In-video questions** that pop up during or at the end of a video to check your learning
* **Readings** to introduce new ideas and build on the concepts from the videos
* **Discussion forums** to discuss, explore, and reinforce new ideas for better learning
* **Discussion prompts** to promote thinking and engagement in the discussion forums
* **Qwiklabs** to introduce real-world, on-the-job situations, and the tools and tasks to complete assignments
* **Practice quizzes** to prepare you for graded quizzes
* **Hands-on activities** toreinforce learned skills for the graded quizzes
* **Graded quizzes** to measure your progress and give you valuable feedback

Hands-on activities promote additional opportunity to build your skills, so try to get as much out of them as possible. Assessments are based on the approach taken by the course to offer a wide variety of learning materials and activities that reinforce important skills. Graded and ungraded quizzes will help the content sink in and reinforce important skills. Ungraded practice quizzes are a chance for you to prepare for the graded quizzes, and both the graded and ungraded quizzes can be taken more than one time.

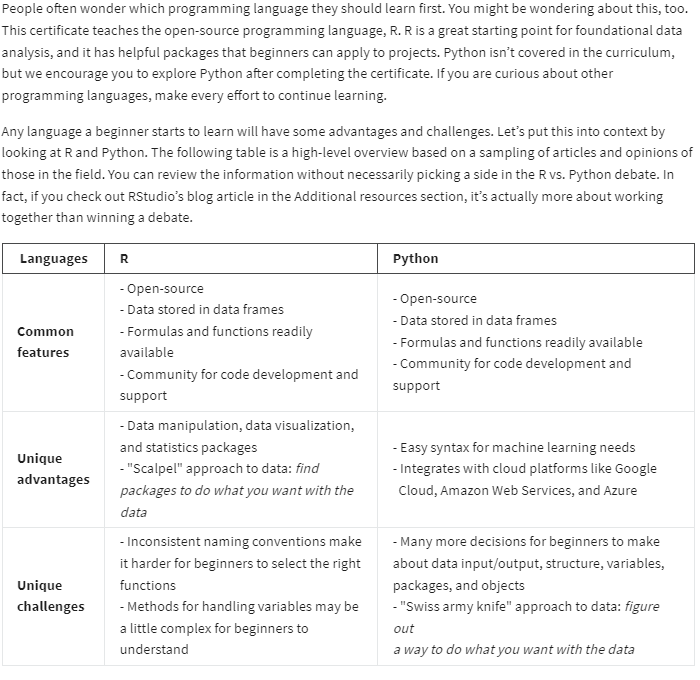
As a quick reminder, this course is designed for all types of learners, so no degree or prior experience is required. Everyone learns differently, and the Google Data Analytics Certificate has been designed with that in mind. Personalized deadlines are just a guide, so feel free to work at your own pace. There is no penalty for late assignments. If you prefer, you can extend your deadlines by returning to **Overview** in the navigation pane and clicking **Switch Sessions**. If you already missed previous deadlines, click **Reset my deadlines** instead.

If you would like to review previous content or get a sneak peek of upcoming content, you can use the navigation links at the top of this page to go to another course in the program. When you pass all the required assignments, you will be on track to earn your certificate.

## **Tips**

* Try to complete all the activities in order, since new information always builds on previous lessons.
* Treat every task as if it is real-world experience. Have a mindset that you are working at a company or in an organization as a data analyst. This will help you apply what you learn in this program to the real world.
* Repeat demonstrated tasks on your own for extra practice and speed. For example, after you follow along with a video once or twice to perform the demonstrated tasks, try performing the same tasks without playing the video and receiving help from the instructor’s prompts.
* Even though they aren’t graded, be sure to participate in and complete all of the practice activities. They will help you build a strong foundation as a data analyst and prepare you for the graded assessments.
* Take advantage of all the additional resources provided, including discussion forums and links to external articles for more information.
* When you encounter useful links in the course, remember to bookmark them so you can refer to the information for study or review.
* Additional resources are free, but some sites place limits on how many articles you can access for free each month. Sometimes you can register on the site for full access, but you can always bookmark a resource and come back to view it later.
* Maximize the value of hands-on activities. Hands-on activities supplement the demonstrated tasks by encouraging additional practice with similar scenarios. A programming language’s syntax will become more natural to you the more you practice using it.
* Create a notebook or document to keep track of things to remember about the R syntax. This will become a handy and personalized reference that you can use throughout the rest of the program and anytime later.

[The R-versus-Python debate](https://www.coursera.org/learn/data-analysis-r/supplement/pGHK8/the-r-versus-python-debate)



## 

## **Additional resources**

For more information on comparing R and Python, refer to these resources:

* [R versus Python, a comprehensive guide for data professionals](https://medium.com/analytics-and-data/r-vs-python-a-comprehensive-guide-for-data-professionals-321e8dead598): This article is written by a data professional with extensive experience using both languages and provides a detailed comparison.
* [R versus Python, an objective comparison](https://www.dataquest.io/blog/python-vs-r/): This article provides a comparison of the languages using examples of code use.
* [R versus Python: What’s the best language for data science?](https://blog.rstudio.com/2019/12/17/r-vs-python-what-s-the-best-for-language-for-data-science/): This blog article provides RStudio’s perspective on the R vs. Python debate.

## **Key takeaways**

Certain aspects make some programming languages easier to learn than others. But, that doesn’t make the harder languages impossible for beginners to learn. On the flip side, a programming language’s popularity doesn’t always make it the best language for beginners either.

R has been used by professionals who have a statistical or research-oriented approach to solving problems; among them are scientists, statisticians, and engineers. Python has been used by professionals looking for solutions in the data itself, those who must heavily mine data for answers; among them are data scientists, machine learning specialists, and software developers.

As you grow as a data analytics professional, you may need to learn additional programming languages. The skills and competencies you learn from your first programming experience are a good foundation. That's why this course focuses on the basics of R. You can develop the right perspective, that programming languages play an important part in the data analysis process no matter what job title you have.

The good news is that many of the concepts and coding principles that you will learn from using R in this course are transferable to other programming languages. You will also learn how to write R code in an Integrated Development Environment (IDE) called RStudio. RStudio allows you to manage projects that use R or Python, or even a combination of the two. Refer to [RStudio: A Single Home for R & Python](https://www.rstudio.com/solutions/r-and-python/) for more information. So, after you have worked with R and RStudio, learning Python or another programming language in the future will be more intuitive.

For a better idea of popular programming languages by job role, refer to [Ways to learn about programming](https://www.coursera.org/learn/data-analysis-r/supplement/y8zTf/ways-to-learn-about-programming). The programming languages most commonly used by data analysts, web designers, mobile and web application developers, and game developers are listed, along with links to resources to help you start learning more about those languages.

Python is more object-oriented, and R is more functional.

[Learning Log: Get ready to explore R](https://www.coursera.org/learn/data-analysis-r/supplement/nFwJ3/learning-log-get-ready-to-explore-r)



## **Overview**

****

So far, you’ve learned about data management, analysis, and visualization. In this upcoming course, you will learn the fundamentals of R and how it can be used to perform the same data analysis steps you already know. Now, you’ll complete an entry in your learning log considering R as a tool for data analysis. By the time you complete this entry, you will be ready to start learning this new and exciting tool!

## **R you ready?**

****

Before you start writing your learning log entry in the template linked below, let’s discuss what exactly R is. **R** is a programming language used for statistical analysis, visualization, and other data analysis. As a data analyst, you will use R to complete many of the tasks associated with the data analysis process. Understanding how it works and why you use it is crucial to developing a mastery of data analytics.

Like the other tools you have already learned in this program, R will be an important part of your data analysis toolkit. You don’t need any previous experience with R for this course; you’ll get a chance to learn the basics and practice writing R code yourself. Then, you can even try using R for your capstone project later!

In the learning log template below, you’ll get a chance to reflect on learning R and what you’re most excited about for this course.



### **Access your learning log**

To use the template for this course item, click the link below and select “Use Template.”

Link to learning log template: [Get ready to explore R](https://docs.google.com/document/d/1r4L9UMxdsxdc24SrFws7dNNmM_7XDDbQ8m0wpwuGIt4/template/preview)

OR

If you don’t have a Google account, you can download the template directly from the attachment below.

[Learning Log Template\_ Get ready to explore R](https://d3c33hcgiwev3.cloudfront.net/tGrlU5-XTx6q5VOfly8eYw_fceedbab923c44b7889850a8e25412f1_Learning-Log-Template_-Get-ready-to-explore-R.docx?Expires=1717545600&Signature=LWpikMA4np-zVO7~SwKojgdjPYi~drqU5hIln4Sjv31HocRhXHqCwn1KV9dIAmOLp8ST1RiPL3HQqvQ51Iafw3iFyIxpQ0QGMTn3obcQknjomoEkHCv1XVMI-WzWNS-KllllrJ3T32mXLi0uw9OVnTSzOByw97MWX2R2TR-wGLc_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[](https://d3c33hcgiwev3.cloudfront.net/tGrlU5-XTx6q5VOfly8eYw_fceedbab923c44b7889850a8e25412f1_Learning-Log-Template_-Get-ready-to-explore-R.docx?Expires=1717545600&Signature=LWpikMA4np-zVO7~SwKojgdjPYi~drqU5hIln4Sjv31HocRhXHqCwn1KV9dIAmOLp8ST1RiPL3HQqvQ51Iafw3iFyIxpQ0QGMTn3obcQknjomoEkHCv1XVMI-WzWNS-KllllrJ3T32mXLi0uw9OVnTSzOByw97MWX2R2TR-wGLc_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

## **Reflection**

****

Take a moment to reflect on learning about R. In your learning log template, write 2-3 sentences (40-60 words) in response to each question below:

* Which parts of R are you excited to learn about?
* Which parts seem most challenging?

When you’ve finished your entry in the learning log template, make sure to save the document so your response is somewhere accessible. This will help you continue applying data analysis to your everyday life. You will also be able to track your progress and growth as a data analyst.

[Fun with R](https://www.coursera.org/learn/data-analysis-r/lecture/xsTG8/fun-with-r)

Hi, great to see you again. When I first learned R, it was the visuals that really got me hooked. I still think it's so cool that you can write a little bit of code, press a button and, presto, out pops an awesome data visualization. Before we get into all the details, I thought it would be fun to give you a quick sneak peek and show you what R can do.

What follows will be a preview of what you're going to learn. By the end of this course, you'll not only understand all this code, you'll be able to write and execute it as well. For now, just sit back, relax and enjoy the show.

Let's start by loading a library and getting a dataset to work with. We can use the Palmer penguins dataset, which contains size measurements for three penguin species that live on the Palmer Archipelago in Antarctica.

This includes data on stuff like body mass, flipper length and bill length. The dataset has 344 rows of information sorted into eight columns.

The Palmer penguins data is popular with analysts and is great for fun exploration, visualization and teaching concepts. We'll see more of this data set later on in the course.

Let's say we want to visualize the relationship between body mass and flipper length. You may guess the larger the penguin, the longer the flipper. We can find out for sure by creating a plot. Let's make a scatter plot. A scatter plot uses points to display the relationship between two variables. So the two variables were going to compare our body mass and flipper length. No need to memorize all these details right now. You'll have time to learn more about them later on.

Let's check out the parts of this code and how they fit together. The first function starts the plot. If we run the code at this point, all we get is a blank plot.

If we add some more code, R will put labels on each access of our plot and add lines for data. Body mass is on the y-axis and flipper length is on the x-axis, but the data points are not yet visible.

To get the complete plot, we can add some more code that tells R how to represent our data. For example, we could use points, bars or lines. We'll use points to create a scatter plot.

We can go further. For example, we can change how the plot looks. Let's change the color of all of the points to purple. You can hit the Up arrow to pull up the last piece of code you ran, so we'll do that now. And then we'll add in color equals purple inside geom point. Now we can hit Enter to run this.

We can also add new information to the plot and use color to highlight it. Let's tell R to assign a different color to each species of penguin. This way we can link data points to each group of penguins.

Gentoos are the largest. The legend just to the right of the plot shows us that the blue points refer to the Gentoos. R automatically creates a legend for the plot to help us understand the color-coding. R does everything you tell it to do and even does stuff you don't ask for. It's just that helpful. We can also use shape to highlight the different penguin species.

Or we can use both color and shape.

In addition to highlighting our data, we can also reorganize it. We can break our data down into smaller groups or subsets and create a plot for each subset. Let's say we want to focus on the data for each species. Facet functions let us create a separate plot for each species. Check this out. Facets are so great. We can even put text on our plot to point to specific data or communicate a message. Let's give our plot a title to clearly indicate its purpose.

Finally, we can save our plot, so we can access or share it later on.

Now, if we click on the Files tab, we'll find our file in the list.

Let's open it up.

Well, that's the end of the show. I hope you enjoyed it as much as I did. We were able to take a big dataset and quickly visualize some significant patterns. These are some of the basic functions in R. In other words, this is just the beginning. It's exciting to think of all the ways R can help you realize the full power of data analysis. As you move forward, you learn more about each of the functions we use to create our plots. By the end of this course, you'll be the one writing and executing all of this code. Coming up, we'll learn more about computer programming and how it can help you analyze your data. See you soon.

[Carrie: Getting started with R](https://www.coursera.org/learn/data-analysis-r/lecture/sqm2J/carrie-getting-started-with-r)

The advice I would give to someone just learning R is that mistakes are part of the process. Errors and error messages are part of the process. When I think about the people who are even better than I am in R, I've come to realize they're not necessarily smarter than I am, but they may be a little bit more persistent and delving a little bit deeper.

Certainly compared to when I started, initially I'd see an error message and think, "I did it wrong, uh-oh, game over. " Now it's like, "That's just part of the game." When I started to get a little bit of exposure to what R looked like, I was like, "That seems too sophisticated. It seems like that probably is really hard." But the people who used it that I had met, we're always **really enthusiastic about it** and they felt like it had so many advantages over other software that you can use for running analyses.

There were a lot of times before I used R where I might use spreadsheets or some other tool and I would be trying to hack at what really needed to happen. Sometimes I was using multiple tools because an individual tool couldn't really do all what I wanted it to do. But it's like I knew in my mind and yet it wasn't totally fluid, the execution of it.

**The more exposure I've gotten to R, the more I realize a lot of what I would try to do that way, I can just do within one program, and it can all interlock really fluidly.** At first, I was really unconfident. I had a couple of scripts where I had some friends who were better at R, people I worked with who would sit down and help me go through and understand the code and so it felt really silly to ask them the simple question of like, **"Okay, but why is a bracket here?" Or "Why would we do this?"** But they were fortunately really patient people.

Then at some point, our entire department said, really everybody needs to be using this because we need everyone on the same platform. We need consistency in our analyses. We need to be able to code review each other's analyses as well. We all took an online course together and that helped me feel really a lot more confident because it was walking through each step of what you needed to know, got an opportunity to practice, and then it felt like, "Okay, even if there's things I don't know, "I've made it through introduction, like I've made it through this next module so I do know something." Then once I started to apply it in my work, there would still be points where I was like, "Wait, I don't know how to solve this problem." Then I would talk to a friend, Google something and generally, I knew a lot more than I thought that I did and from that, I suddenly unlocked my ability to produce a whole lot of analyses quickly with the big dataset and also produce a whole lot of data visualizations really quickly using ggplot2. Hi, my name's Carrie and I'm a Research Manager within People Operations at Google.

[Meet and greet](https://www.coursera.org/learn/data-analysis-r/discussionPrompt/HHB6q/meet-and-greet)

**1- What is your professional background?** USAF veteran, Civil Engineer(Civilian), Dental Lab.Tech(USAF).

**2- What led you to enroll in this course?** TI was something I wanted to engage in to learn and work with after finishing my graduation at Civil Engineering. It gets more and more interesting with every line of code and classes.

**3- What programming languages have you worked with?** Swift, CSS/HTML, JavaScript, Ruby, Python << Beginner level to intermediate on all those languages.

**4- What personal or professional projects have you used programming for?** I have used programming to do websites and mobile applications.

**5- What do you enjoy most about programming?** The ability to solve problems, make life easier for humanity and create great things with code!

**6- If you are new to programming, what are your feelings about learning how to program using R—are you excited, a little nervous, or both?** I am not new to programming and I am very excited to start learning a new language and using a new IDE like RStudio!

PROGRAMMING AS A DATA ANALYST

[Programming languages](https://www.coursera.org/learn/data-analysis-r/lecture/hoWfr/programming-languages)

[Ways to learn about programming](https://www.coursera.org/learn/data-analysis-r/supplement/y8zTf/ways-to-learn-about-programming)

[From spreadsheets to SQL to R](https://www.coursera.org/learn/data-analysis-r/supplement/jNOkl/from-spreadsheets-to-sql-to-r)

Although the programming language R might be new to you, it actually has a lot of similarities to the other tools you have explored in this program. In this reading, you will compare spreadsheet programs, SQL, and R to have a better sense of how to use each moving forward.

## **Spreadsheets, SQL, and R: a comparison**

As a data analyst, there is a good chance you will work with SQL, R, and spreadsheets at some point in your career. Each tool has its own strengths and weaknesses, but they all make the data analysis process smoother and more efficient. There are two main things that all three have in common:

* **They all use filters:** for example, you can easily filter a dataset using any of these tools. In R, you can use the filter function. This performs the same task as a basic SELECT-FROM-WHERE SQL query. In a spreadsheet, you can create a filter using the menu options.
* **They all use functions:** In spreadsheets, you use functions in formulas, and in SQL, you include them in queries. In R, you will use functions in the code that is part of your analysis.

The table below presents key questions to explore a few more ways that these tools compare to each other. You can use this as a general guide as you begin to navigate R.



[Introduction to R](https://www.coursera.org/learn/data-analysis-r/lecture/hYFgv/introduction-to-r)

Now that we've talked about programming languages in general, let's get to know R. So what is R? R is a programming language frequently used for statistical analysis, visualization and other data analysis. Later on, you'll take a tour of Rstudio, which is a popular software environment for the R language. In this video, we'll discuss R's main features and functions and its advantages for data analysis. R is super cool.

R is based on another programming language named S. In the 1970s, John Chambers created S for internal use at Bell Labs, a famous scientific research facility. In the 1990s, Ross Oaxaca and Robert Gentleman developed R at the University of Auckland, New Zealand.

The title R refers to the first names of its two authors and plays on a single- letter title of its predecessor S. Since then, R has become a preferred programming language of scientists, statisticians and data analysts around the world. There's lots of reasons why people who work with data love R. I want to share four with you.

**R is accessible, data-centric, open-source and has an active community of users**. First R is an accessible language for beginners. Lots of people without a traditional programming language learn R. I should know. I'm one of them. R really appeals to anyone who wants to solve problems that involve data. And that's one of the things that's so great about R. It's all about data.

R is what's known as a data-centric programming language. It's specifically designed to make data analysis easier, more efficient and more powerful.

Another awesome thing about R is that it's open source. Open source means that the code is freely available and may be modified and shared by the people who use it. Let's pause for a moment and unpack how amazing this is. First, anyone can use R for free. Second, anyone can modify the code, fix bugs and improve it. In fact, over the years, lots of excellent programmers have made improvements and fixes to the R code. For example, anyone who knows the R language can create what's called an add-on package. We'll talk more about R packages later. For now, just know that literally thousands of R packages exist, and they were all built by people who wanted to solve specific problems. A lot of these packages are super useful for data analysts. As an R user, you now enjoy the benefit of the shared knowledge.

And let me just add, the R community is the best. This vibrant, diverse and accessible community is so supportive of new learners. You can go online anytime to find answers to all your R questions. Check out websites like R for Data Science Online Learning Community and RStudio Community. On top of that, R users are all over Twitter and other social media. You'll discover tons of resources for professional networking, mentoring and learning.

Now that we know more about the general benefits of R, let's talk about some specific situations when you might use it for data analysis. Here's three scenarios: reproducing your analysis, processing lots of data, and creating data visualizations. First R can save and reproduce every step of your analysis. Earlier, we discussed how data analysis is most useful when you can easily reproduce your work and share it with others. In R, reproducing your analysis is as easy as pressing a button on your keyboard. Your code stores it forever. And you can share it with anyone at any time.

Processing lots of data is also something R does really well, just like SQL. As you learned earlier spreadsheets organize projects in sheets or tabs. If you've ever had to deal with spreadsheet files that have tons of sheets or lots of data in each sheet, you know that things can start to move very slowly. Working with too much data in a spreadsheet can even cause crashes. R can handle large amounts of data much more quickly and efficiently. Finally R can create powerful visuals and has state-of-the-art graphic capabilities. As you've seen in this program, tools like spreadsheets and Tableau offer lots of options for visualizing your data. R's on another level. With only a small bit of code, you can create histograms, scatter plots, line plots and so much more. And that's just the beginning. If you work with more advanced packages, you can make some seriously impressive data visualizations.

Learning R is a huge benefit to anyone interested in becoming a data analyst. As I mentioned earlier, knowledge of R will help you stand out as a job candidate. And as you keep moving forward, R will help you find solutions for more complex data problems. You can keep learning about R throughout your career as a data analyst. The sky's the limit when it comes to developing your data analysis skills. That's all for now.

Coming up, we'll check out the RStudio environment together. Before you use RStudio, you need to download and install the basic R interface. You'll learn how to do that in an upcoming reading. Most analysts who work with the R language use the RStudio environment to interact with R, and not the basic interface. That's why we're focusing on RStudio in this program. Following this video, you'll find resources for downloading R and RStudio if you're interested in learning more. Bye for now.

[Optional Hands-On Activity: Downloading and installing R](https://www.coursera.org/learn/data-analysis-r/quiz/mWN3I/optional-hands-on-activity-downloading-and-installing-r)

### 1.

Question 1



## **Activity overview**

****

Earlier in this course, you learned about R, a programming language used for statistical analysis, visualization, and other data analysis. In this activity, you’ll complete the steps to download and install R on your computer.

By the time you complete this activity, you will be able to use R without internet access and independent of the RStudio cloud-based suite. This will enable you to use R with more flexibility, which is important for programming effectively during your career as a data analyst.

## **Prepare for installation**

****

* Note: This is an optional activity. RStudio Cloud (which has transitioned to the company *Posit Cloud*) is the primary tool you will use for this course, but you can also install R to your computer for offline use. Please keep in mind that Chrome OS does not support the installation of R. If you are completing this course on a Chromebook, you should skip this activity or refer to the Linux workaround linked below.

In order to get started, you need to know your operating system. Your operating system (OS) is the firmware that makes up your computer’s main interface. Some common OS’s include MacOS (Apple), Windows OS (Microsoft), Chrome OS (Google). The OS on your device determines which version of R you will install.

* Note: If you use Chrome OS, you will need to enable Linux (Beta) in order to use R. [This guide](https://blog.sellorm.com/2018/12/20/installing-r-and-rstudio-on-a-chromebook/) details how to install R on a Chromebook. Otherwise, you can use an online coding platform like RStudio Cloud or Kaggle.

Once you have determined your OS and the version of R it requires, it is time to download and install its assets.

## **Download R**

****

1. Go to the R website and navigate to the [download page](https://cran.r-project.org/mirrors.html) on the Comprehensive R Archive Network. The download page brings you to a list of locations to download R.

2. Click one of the “mirrors,” or download locations. This will bring you to a page with download links corresponding to each OS. Don’t worry about which mirror to pick--all of them host the same R installation files.

3. Find your OS, click its corresponding link, and download the base package. The description should say “Binaries for base distribution.”

4. Click the download link to begin downloading R.

## **Install R**

****

1. Once your download is complete, open the downloaded file. This will open R.

2. Select your preferred language from the drop-down menu. Then, click Next >.

3. Review the license information for R for your OS. This describes its open-source availability, which means it may be modified and shared by the people who use it. Click Next >.

4. Choose the install location for R. To pick an install location, click Browse and navigate to the folder you’d like to select. If you are not picky about where you want to install these files, the default location provided will be fine. Click Next >.

5. Click the checkboxes for the appropriate files you need. For example, if you have a 64-bit system, only download those files. Click Next >.

6. Select No for customizing your startup options. Click Next >. Then at the following screen, click Next >. You have now installed R to your computer.

## **Using R**

****

1. Open R and locate the R Console. This is a window in which you can write and execute commands in R. Find the > symbol at the bottom of the console and click the empty space to the right of it.

2. Enter a simple display command for your first command. Type print(“Hello world!”) into the command prompt. Press Enter (Windows) or Return (Mac) to show the result: [1] "Hello World!" Note that whenever you execute a command, R will give a number to each line of output that results.

3. Enter a simple mathematical equation for your second command. Type 1+2 into the command prompt. Press Enter (Windows) or Return (Mac) to receive the answer, which is 3. Later in this course, you will practice more simple math in R.

4. Enter a quit command for your last command. Type q() into the prompt and press Enter (Windows) or Return (Mac). The program will close.

## **Reflection**

****

In this activity, you downloaded and installed files for the R programming language. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

* What is an advantage of installing R instead of using it on an online platform?
* How will learning R help you build your data analytics skills?

[Optional Hands-On Activity: R Console](https://www.coursera.org/learn/data-analysis-r/quiz/rP1nm/optional-hands-on-activity-r-console)

## **Activity overview**

****

In the last activity, you downloaded and installed R. You can use the R environment and programming language to conduct data analysis and create visualizations. In this activity, you'll review the basics of working with the R Console and learn how to write and execute a basic command.

This will enable you to better understand the standard R interface. While you will use RStudio for most of the activities in this course, it is useful to know the basics of a programming interface as this will likely come up in your day-to-day work as a data analyst.

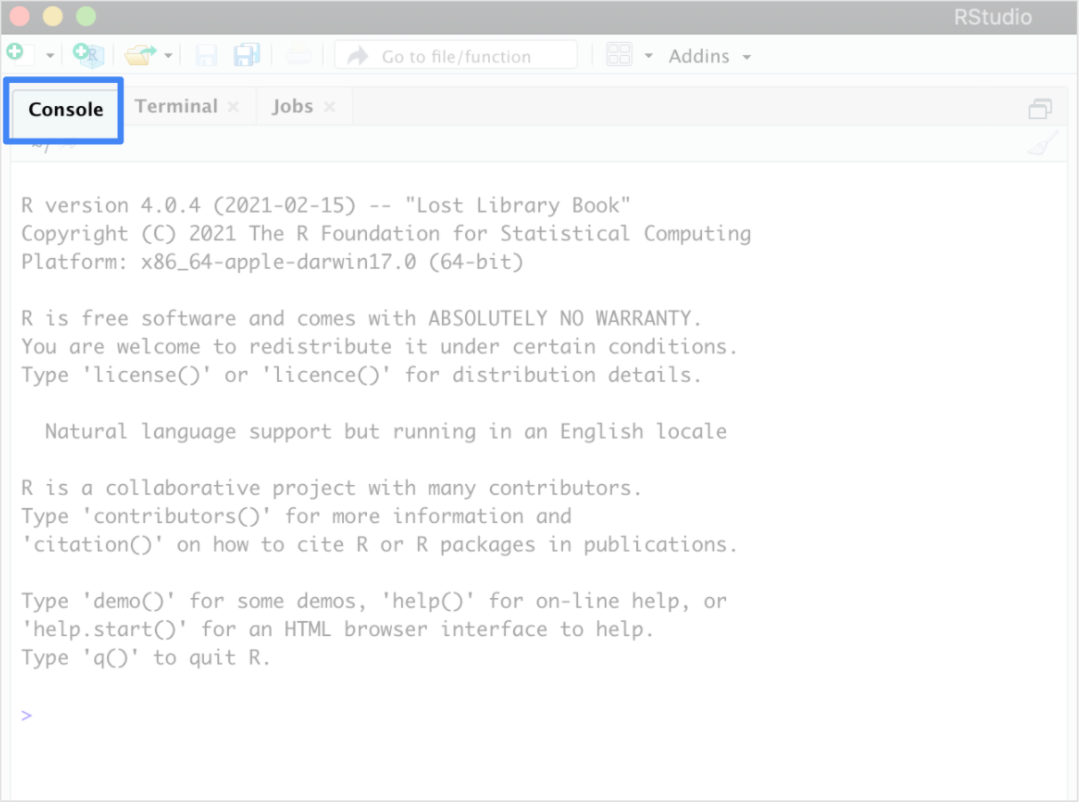
## **What is the R Console?**

****

* Note: This is an optional activity. RStudio Cloud is the primary tool you will use for this course, but you can also install R to your computer for offline use. Please keep in mind that Chrome OS does not support the installation of R. If you are completing this course on a Chromebook, you should skip this activity or refer to the Linux workaround linked below.

The R Console is the program window in R where you make use of the R programming language. It is an interface that lets you view, write, edit, and execute your R code.

Programs like RStudio, an interactive development environment (IDE) for programming in R, use the R Console and other tools to make it easier to write and execute R code. In RStudio, the R Console is often referred to as the console pane (pictured below). It lets you perform any tasks you’d do in the R Console.

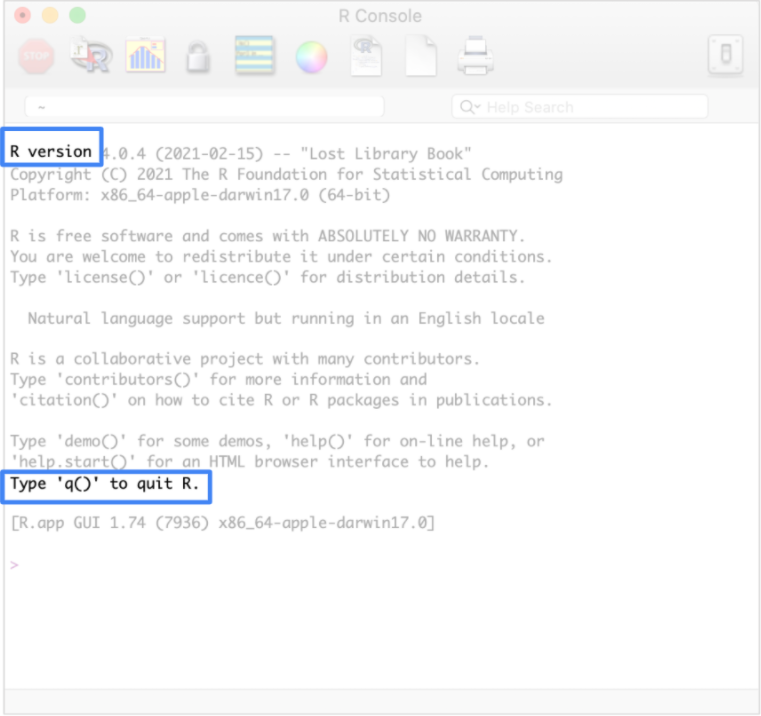


However, as you start coding in R, it’s helpful to begin with the simplicity of just the R Console. During this hands-on activity, you’ll use the R Console to perform simple mathematical operations.

## **Use the R Console**

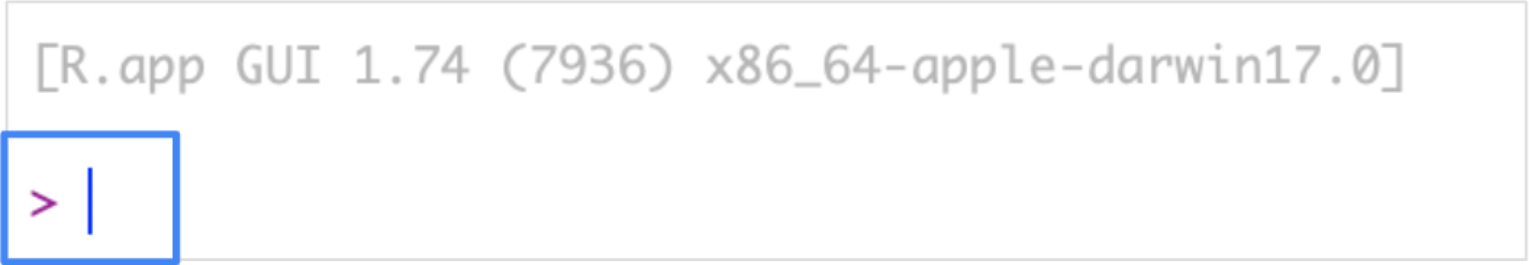
****

1. Open the R program to use the R Console on your computer. You will find that the console populates a default message. The message starts with R version and your version number, and ends with Type ‘q()’ to quit R. Above the message, you will find a menu with icons that represent the functions of the console and graphical user interface (known in the program as RGui).



2. Click in the blank space to the right of the > symbol at the bottom of the console.

This is the prompt, and anything you type after it will be read as executable R code when you press Enter (Windows) or Return (Mac). Keep in mind that everything you write in the R Console disappears after you end your session (or close the console). If you want to save the code you execute, it is better to save it in a text file or an .rmd file (which you will learn more about in upcoming lessons).



3. Type citation() after the prompt and press Enter (Windows) or Return (Mac). This returns instructions for how to cite R in a publication. You don’t need to worry about this now, but it will be helpful if you ever use R in a research paper or article.

After you execute the line, the > prompt will generate again and you will be able to write a new line of R code. Now, write a mathematical operation. Start with simple addition by using the plus operator (+).

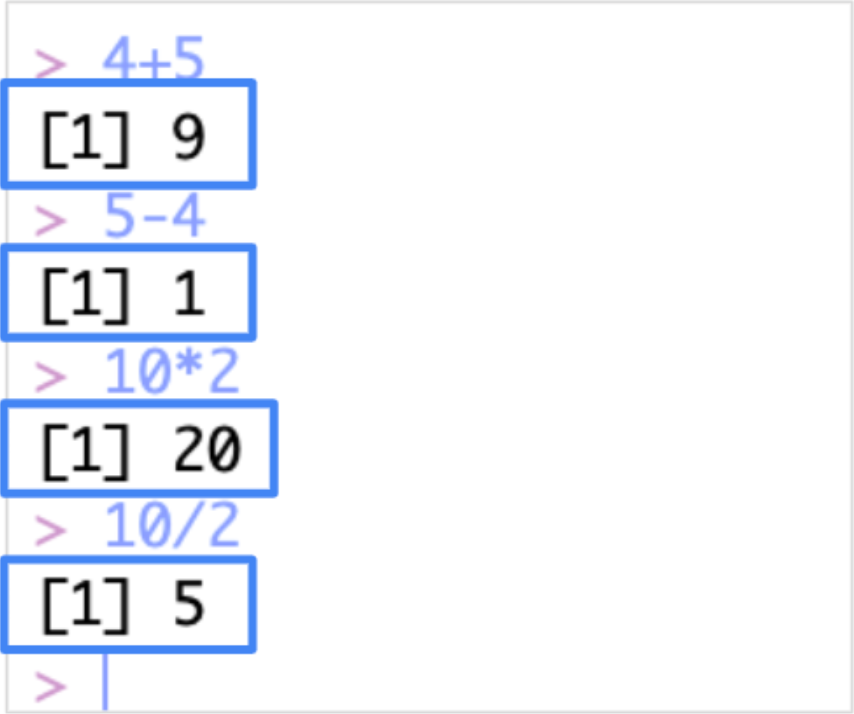
4. Type 4, then a +, then the number 5. The text you type should look like: 4+5. Press Enter (Windows) or Return (Mac). The R Console will return the answer to this question, which is 9.

5. On a new line, type 5-4 to use the subtraction operator (-). Press Enter (Windows) or Return (Mac) to execute the code and return the answer, which is 1.

6. On a new line, use the multiplication operator (\*) to multiply two numbers. Type 10\*2 and then press Enter (Windows) or Return (Mac). This will execute the code and return the answer, which is 20.

7. On a new line, use the division (/) operator to divide two numbers. Type 10/2 and then press Enter (Windows) or Return (Mac). This will execute the code and return the answer, which is 5.

Your R code and results should look like this:



Congratulations, you’ve written code in R! You can use R to complete mathematical operations, among many other useful data analysis tasks. This is just the beginning of your journey with writing in R.

## 

## **Reflection**

****

In this activity, you used the R console to write some basic functions. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

* What does the R console teach you about programming in the R interface?
* What is the difference between using the R console versus writing R code in a text file?

LEARN PROGRAMMING USING RStudio

[Intro to RStudio](https://www.coursera.org/learn/data-analysis-r/lecture/HuGrI/intro-to-rstudio)

It's time to take our tour of RStudio. The examples we'll look at are from RStudio Cloud, but RStudio works in a similar way across all platforms. Feel free to use the platform that works best for you.

RStudio's an IDE or integrated development environment. This means that RStudio brings together all the tools you might want to use in a single place. The R console which we explored earlier is one part of this environment. RStudio also includes an editor for writing code, and tools for managing your data and creating visuals. RStudio is built specifically for use with R. It'll help maximize your productivity as a data analyst. Data analysis is like driving a car. You can think of R and RStudio as different parts of this car. R is like a car engine. RStudio is kind of like the accelerator, the steering wheel, and dashboard all-in-one. It lets you tell the engine what to do and helps you get to where you want to go. Just as a speedometer and navigation system make driving much easier, RStudio's environment makes using R much easier. In an earlier reading, you learned how to access RStudio. So let's log into RStudio now and explore. The RStudio environment has four main windows called panes. Each pane helps you perform different functions. The first time you open RStudio, you'll see three panes. A fourth pane is hidden by default, but it's easy to open. Just click on File in the menu, then select New File and R Script. RStudio has lots of keyboard shortcuts. To learn more check out Keyboard Shortcuts Help.

You can make the panes smaller or larger by clicking on the minimize or maximize buttons at the upper right of each pane.

You can also click and drag the borders of the panes to adjust their size.

Click on the Panes button for more feature options.

Now that we've got all four panes open, let's explore each of them. We'll start on the lower left and move clockwise from there. You might recognize the R console from an earlier reading. As a quick refresher, the console is the place where you give commands to R. For example, we can tell R to show us a summary of the penguins data that we used in an earlier video to create visuals. You'll need to install and load the palmer penguins dataset if you haven't done so already.

Above the console in the upper left is the source editor pane. You'll use the source editor when working with R Scripts. There are two main ways of writing code in RStudio: using the console or using the source editor. You can type commands directly into the console, but they'll be forgotten when you close your current session. As we've discussed, it's important to be able to reproduce and share all the steps of your analysis. If you save your script in the editor, you can access your work again at any time and show others how you did it.

The source editor and the console also work together in RStudio. When you execute code in the editor, the code automatically appears in the console. If you're working on a long analysis, this makes it easy to execute the entire code all at once or run specific sections of it as you go along. Let's run some code in the editor and check it out.

Pro tip: Always keep in mind that R is case-sensitive. Here we use a capital V for the View function.

Next, let's go to the Environment pane in the upper right. Here you'll find all the data you currently have loaded and can easily organize and save it. For example if you import data from a spreadsheet, it'll be visible in the Environment pane. You can view each object in the Environment pane by clicking on it. You can also toggle between a List view and a Grid view.

To the right of the Environment tab, you'll find the History tab. All your previous commands are saved here and they're easy to search and re-execute. You'll find the most recent line of code at the bottom of the list. You can copy any line to the command console by double-clicking it. In the lower right, you'll see a pane that has tabs for Files, Plots, Packages, and Help. The Files tab gives you access to your file directory and shows the contents of the current working folder. You can easily find and manage all your files and create new project folders. Next is the Plots tab. If we create a plot, the result appears here. For example, we can create a scatter plot with the penguins dataset we used earlier.

You'll learn more about creating plots in RStudio later on. Earlier, we talked about R packages which are custom solutions to data problems developed by R users. RStudio gives you access to a library of R packages known as the tidyverse. You can upgrade, install, and manage your library in the Packages pane. Packages loaded in your current session have a check mark. Later on, we'll explore the tidyverse in more detail. Finally, click on the Help tab. Here you can find helpful resources for R and RStudio. There are tons of resources out there to help answer all your questions. Be sure to take advantage of them. That's our tour of RStudio. We're just scratching the surface of what RStudio can do. Soon you'll get to explore RStudio in more detail. Speaking as a data professional, I love working in RStudio. It makes my work so much easier, faster, and better. Congratulations on finishing another step in your data analyst learning journey. Coming up, we'll learn some basic programming concepts. Then we'll start working with R. For those of you who are new to programming, you're about to write your first lines of code. See you then.

[Hands-On Activity: Cloud access to RStudio](https://www.coursera.org/learn/data-analysis-r/quiz/kAQV5/hands-on-activity-cloud-access-to-rstudio)

[Optional Hands-On Activity: Get started in RStudio Desktop](https://www.coursera.org/learn/data-analysis-r/quiz/1eqMt/optional-hands-on-activity-get-started-in-rstudio-desktop)

[When to use RStudio](https://www.coursera.org/learn/data-analysis-r/supplement/Q1J3x/when-to-use-rstudio)

[R&R...Studio!](https://www.coursera.org/learn/data-analysis-r/discussionPrompt/RmWTf/r-r-studio)

[Connecting with other analysts in the R community](https://www.coursera.org/learn/data-analysis-r/supplement/XgjMC/connecting-with-other-analysts-in-the-r-community)

[Test your knowledge on programming with RStudio](https://www.coursera.org/learn/data-analysis-r/quiz/3uM4T/test-your-knowledge-on-programming-with-rstudio)

M1 CHALLENGE

[Glossary: Terms and definitions](https://www.coursera.org/learn/data-analysis-r/supplement/EUfkW/glossary-terms-and-definitions)

[Module 1 challenge](https://www.coursera.org/learn/data-analysis-r/exam/LuT1B/module-1-challenge)