The R-versus-Python debate

People often wonder which programming language they should learn first. You might be wondering about this, too. This certificate teaches the open-source programming language, R. R is a great starting point for foundational data analysis, and it has helpful packages that beginners can apply to projects. Python isn’t covered in the curriculum, but we encourage you to explore Python after completing the certificate. If you are curious about other programming languages, make every effort to continue learning.

Any language a beginner starts to learn will have some advantages and challenges. Let’s put this into context by looking at R and Python. The following table is a high-level overview based on a sampling of articles and opinions of those in the field. You can review the information without necessarily picking a side in the R vs. Python debate. In fact, if you check out RStudio’s blog article in the Additional resources section, it’s actually more about working together than winning a debate.

| **Languages** | **R** | **Python** |
| --- | --- | --- |
| **Common features** | - Open-source - Data stored in data frames - Formulas and functions readily available - Community for code development and support | - Open-source - Data stored in data frames - Formulas and functions readily available - Community for code development and support |
| **Unique advantages** | - Data manipulation, data visualization, and statistics packages - "Scalpel" approach to data: find packages to do what you want with the data | - Easy syntax for machine learning needs - Integrates with cloud platforms like Google  Cloud, Amazon Web Services, and Azure |
| **Unique challenges** | - Inconsistent naming conventions make it harder for beginners to select the right functions - Methods for handling variables may be a little complex for beginners to understand | - Many more decisions for beginners to make about data input/output, structure, variables, packages, and objects - "Swiss army knife" approach to data: figure out  a way to do what you want with the data |

## 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.

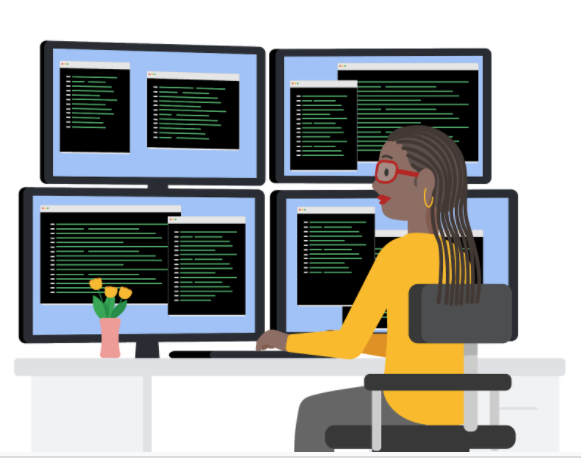
**Access your learning log**

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)

Ways to learn about programming

Writing programming language code can be an exciting and rewarding experience. The programming field has a long history of people helping each other improve their skills and develop best practices. You will focus on the R programming language in this course, but in the future you might choose to pursue additional programming languages based on your interests and professional goals. This reading is a general guide to help you decide which programming languages are best suited for you.



**Popular programming languages by profession**

Let’s go through some potential job titles you might encounter and the most popular programming languages used in those professions. Also included is a list of additional resources for you to explore and learn more about each of the programming languages introduced.

**Data analyst**

A data analyst collects, transforms, and organizes data to draw conclusions, make predictions, and drive informed decision-making. The most popular programming languages used by data analysts are R and Python.

**R** offers convenient statistical features for data analysis and is useful for creating advanced data visualizations. Check out these resources to learn more about R:

* [The R Project for Statistical Computing](https://www.r-project.org/): a website for downloading R, documentation, and help
* [R Manuals](https://cran.r-project.org/manuals.html): links to manuals from the R core team, including introduction, administration, and help
* [Coding Club R Tutorials](https://ourcodingclub.github.io/tutorials.html): a collection of coding tutorials for R
* [R for Beginners](https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf): a starting guide to help you work with data, graphics, and statistics in R

**Python** is a general-purpose language that you can use to create what you need for data analysis. Here are a few resources to begin learning Python:

* [The Python Software Foundation (PSF)](https://www.python.org/about/gettingstarted/): a website with guides to help you get started as a beginner
* [Python Tutorial](https://docs.python.org/3/tutorial/): a Python 3 tutorial from the PSF site
* [Coding Club Python Tutorials](https://ourcodingclub.github.io/tutorials.html): a collection of coding tutorials for Python

**Web designer**

A web designer is responsible for the styling and layout of web pages containing text, graphics, and video. Web designers generally use Hypertext Markup Language v5 (HTML5) and Cascading Style Sheets (CSS) to create web pages.

**HTML5** provides structure for web pages and is used to connect to hosting platforms. Learn more about HTML5 and CSS using these resources:

* [HTML Tutorial](https://www.tutorialrepublic.com/html-tutorial/): an introduction to HTML with links to HTML5 features, examples, and references
* [HTML5 Cheat Sheet](https://www.wpkube.com/html5-cheat-sheet/): a handy summary of HTML5 tags, attributes, and compatibility with HTML4
* [HTML5 and CSS Fundamentals course](https://www.edx.org/course/html5-and-css-fundamentals): a free W3C course on edX; a verified course certificate can be issued for $199

**CSS** is used for web page design and controls graphic elements (color, layout, and font) and page presentation on multiple devices (large screens, mobile screens, and printers). Check out these cheat sheets for CSS:

* [Interactive CSS Cheat Sheet](https://htmlcheatsheet.com/css/): includes the most common CSS snippets for gradient, background, font-family, border, and much more
* [50 Best HTML & CSS Cheat Sheets](https://sharethis.com/best-practices/2020/02/best-html-and-css-cheat-sheets/): a list of 50 cheat sheets–choose a few that are useful to you

**Mobile application developer**

A mobile application developer uses programming to create applications used on laptops, mobile phones, and tablets. The most popular programming languages for mobile application developers are Swift, Java, and C#.

**Swift** (for Apple platforms) is an open source scripting language for macOS, iOS, watchOS, and tvOS. Its main goal is to make applications run faster. Browse these resources for more information about Swift:

* [Swift.org](https://swift.org/about/): an open source community with resources to learn how to use Swift, including videos and sample code
* [Swift developer site](https://developer.apple.com/swift/): an Apple developer website with information for developers who want to use Swift
* [Swift development resources](https://developer.apple.com/swift/resources/): Apple’s collection of documentation, sample code, videos, and recommended books

**Java** (for Android devices) is the official language for Android development. The article [I want to develop Android apps - which languages should I learn?](https://www.androidauthority.com/develop-android-apps-languages-learn-391008/) explores some other languages used for Android development. Check out these resources for Java:

* [Android Studio](https://developer.android.com/studio): a downloadable integrated development environment (IDE) with tools to build apps for Android devices
* [Build your first Android app in Java](https://developer.android.com/codelabs/build-your-first-android-app#1): instructions for installing Android Studio and creating your first app
* [Java tutorial for beginners: write a simple app with no previous experience](https://www.androidauthority.com/java-tutorial-for-beginners-write-a-simple-app-with-no-previous-experience-1121975/): an overview of how to learn Java, with examples

**C#** (pronounced C-sharp) is an object-oriented programming language that is widely used to create mobile apps in the .NET open source developer platform. Xamarin extends the .NET platform with a framework for developers to create cross-platform mobile apps for both iOS and Android. Here are a few resources to help you learn C#:

* [Microsoft .NET learning materials for C#](https://dotnet.microsoft.com/learn/csharp): includes free courses, tutorials, and videos to learn the programming language C#
* [Microsoft Xamarin learning materials](https://dotnet.microsoft.com/learn/xamarin): includes free courses, tutorials, and videos to learn about mobile development with Xamarin
* [Xamarin Tutorial - build your first iOS or Android app in C#](https://dotnet.microsoft.com/learn/xamarin/hello-world-tutorial/intro): instructions for building a mobile app that displays the text “Hello World”
* [Learn C# from Codecademy](https://www.codecademy.com/learn/learn-c-sharp): a website with free basic interactive lessons, and additional activities that can be accessed with a monthly subscription

**Web application developer**

A web application developer designs and develops network applications used across the web. The most popular programming languages used by web application developers are Java, Python, Ruby, and PHP.

**Java** is widely used to create enterprise web applications that can run on multiple clients. Java’s main strength is its “Write Once, Run Anywhere” (WORA) approach.Browse these resources to learn more about Java:

* [Oracle Java Tutorials](https://docs.oracle.com/javase/tutorial/): Java tutorials from Oracle documentation
* [Java for Beginners](https://www.homeandlearn.co.uk/java/java.html): a free Java course for beginners from the website “Home and Learn”

**Python** is a general-purpose programming language. Check out the Python resources listed in the data analyst section.

**Ruby** is a general-purpose, object-oriented programming language used for web application development. Ruby isn't the same as Ruby on Rails, which is an open source web application framework that runs using Ruby. Browse these resources to learn more about Ruby:

* [Ruby news](http://ruby-doc.org/): information about the latest Ruby releases and links to other resources
* [Ruby documentation](http://www.ruby-lang.org/en/documentation/): includes guides, tutorials, and reference material to help you learn more about Ruby
* [Ruby programmer’s guide](http://ruby-doc.com/docs/ProgrammingRuby/): a tutorial and reference guide for Ruby
* [Learn Ruby from Codecademy](https://www.codecademy.com/learn/learn-ruby): a website with free basic interactive lessons, and additional activities that can be accessed with a monthly subscription

**PHP** is a scripting language particularly suited for web application development. It was based on Perl, another programming language. PHP is simple, flexible, and relatively easy to learn. Check out these resources to learn more about PHP:

* [PHP downloads and documentation](https://www.php.net/): information about the latest PHP releases and links to other resources
* [PHP the Right Way](https://phptherightway.com/): a quick reference for popular PHP coding standards
* [Interactive PHP tutorial](https://www.learn-php.org/): a free tutorial that runs PHP code in exercises

**Game developer**

A game developer is an application developer who specializes in video game creation. Game developers most commonly use the programming languages C# and C++.

**C#** is an object-oriented programming language that is widely used to create games. Check out the C# resources listed in the mobile application developer section.

**C++** is an extension of the C programming language that is also used to create console games, like those for Xbox. Browse more information about C++:

* [Microsoft resources for C++](https://docs.microsoft.com/en-us/cpp/?view=msvc-160): learn how to install the Visual Studio IDE and write C++ code
* [Microsoft C++ and C# code samples for gaming](https://docs.microsoft.com/en-us/samples/browse/?languages=cpp&terms=gaming): a resource with over 40 C++ and C# code samples for gaming
* [Interactive C++ tutorial](https://www.learn-cpp.org/): a free tutorial that runs C++ code in exercises

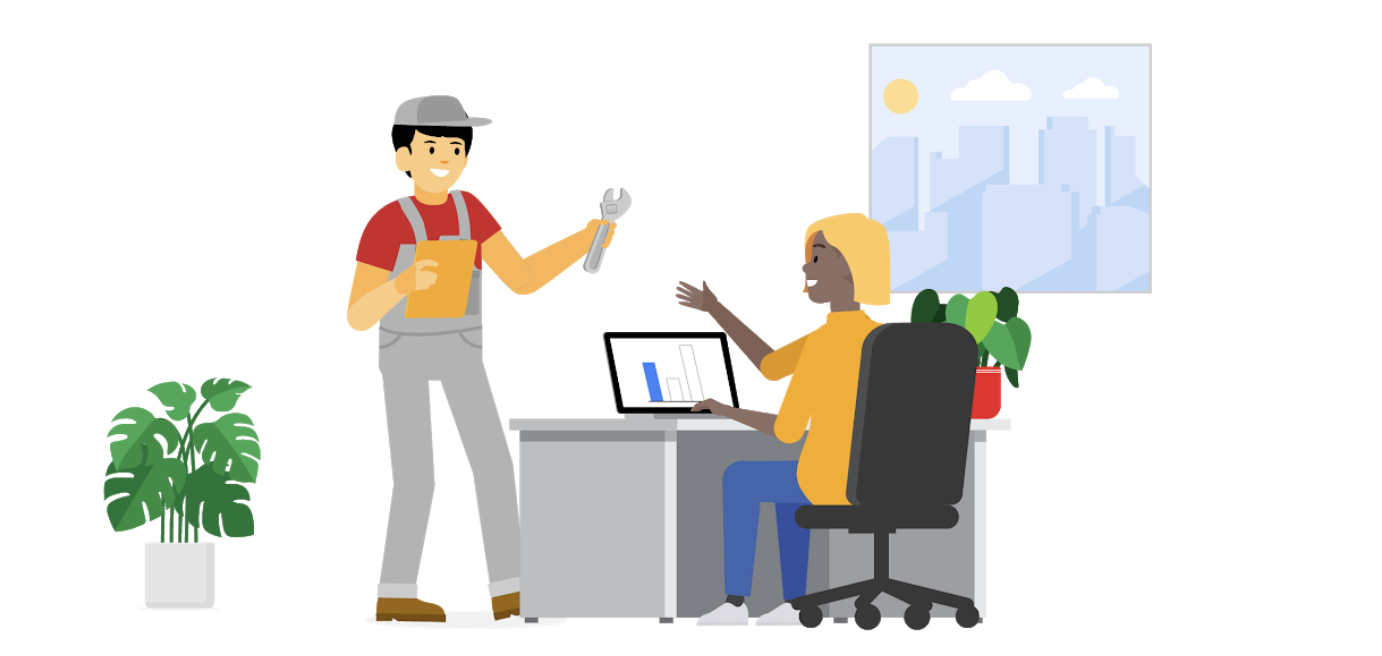
**Tips for learning programming languages**

Here are a few tips to follow when you start learning a new programming language:

* Define a practice project and use the language to help you complete it. This makes the learning process more practical and engaging.
* Keep previous concepts and coding principles in mind. Many of these are transferable between programming languages. So, after you have learned one language, learning a second or third programming language tends to be much easier.
* Create and keep good notes and cheat sheets in whatever format (handwritten or typed) that works best for you.
* Create an online filing system for information that you can easily access while you work in various programming environments.

When to use RStudio

As a data analyst, you will have plenty of tools to work with in each phase of your analysis. Sometimes, you will be able to meet your objectives by working in a spreadsheet program or using SQL with a database. In this reading, you will go through some examples of when working in R and RStudio might be your better option instead.



**Why RStudio?**

One of your core tasks as an analyst will be converting raw data into insights that are accurate, useful, and interesting. That can be tricky to do when the raw data is complex. R and RStudio are designed to handle large data sets, which spreadsheets might not be able to handle as well. RStudio also makes it easy to reproduce your work on different datasets. When you input your code, it's simple to just load a new dataset and run your scripts again. You can also create more detailed visualizations using RStudio.

**When RStudio truly shines**

When the data is spread across multiple categories or groups, it can be challenging to manage your analysis, visualize trends, and build graphics. And the more groups of data that you need to work with, the harder those tasks become. That’s where RStudio comes in.

For example, imagine you are analyzing sales data for every city across an entire country. That is a lot of data from a lot of different groups–in this case, each city has its own group of data.

Here are a few ways RStudio could help in this situation:

* Using RStudio makes it easy to take a specific analysis step and perform it for each group using basic code. In this example, you could calculate the yearly average sales data for every city.
* RStudio also allows for flexible data visualization. You can visualize differences across the cities effectively using plotting features like facets–which you’ll learn more about later on.
* You can also use RStudio to automatically create an output of summary stats—or even your visualized plots—for each group.

As you learn more about R and RStudio moving forward in this program, you’ll get a better understanding of when RStudio should be your data analysis tool of choice.

**For more information**

* [**The Advantages of RStudio**](https://www.theanalysisfactor.com/the-advantages-of-rstudio/): This web page explains some of the reasons why RStudio is many analysts’ preferred choice for interfacing with R. You’ll learn about the advantages of using RStudio for data analysis, from ease of use to accessibility of graphics and more.
* [**Data analysis and R programming**](https://lgatto.github.io/2017_11_09_Rcourse_Jena/before-we-start.html): This online introduction to data analysis and R programming is a good starting point for R and RStudio users. It also includes a list of detailed explanations about the advantages of using R and RStudio. You’ll also find a helpful guide for getting set up with RStudio.

Connecting with other analysts in the R community

R is a powerful tool in your data analysis toolkit–and it also has a powerful community of users who are excited to share, collaborate, and connect with others. This reading will give you a few places where you can start to connect, online and in-person, with other analysts in the R community.



**Online communities**

Online communities allow you to connect with other R users no matter where you live. This list includes forums and discussion channels where you can join the conversation. It also includes social media tags you can use on your existing social media platforms to connect with other data analysts.

* [**RStudio Community:**](https://community.rstudio.com/) The RStudio Community forum is a great place to get help and find solutions to challenges you have with R–and maybe help someone else out, too!
* [**r/RLanguage**](https://www.reddit.com/r/Rlanguage/): The R language subreddit is an active online community on the social media platform Reddit, where R users go to discuss R, ask questions, and share tips.
* [**rOpenSci**](https://discuss.ropensci.org/): rOpenSci has a community forum where R users can ask questions and search for solutions. It also includes links to their Best Practices guide and support pages.
* [**R4DS Online Learning Community and Slack channel:**](https://www.rfordatasci.com/) This is a community with another Slack channel where R learners and mentors can gather and connect. This is a great place to chat about using R for data science.
* [**Twitter #rstats**](https://twitter.com/hashtag/rstats?lang=en): If you use Twitter, you can connect with other R users using the hashtag #rstats; a lot of R developers and analysts are active on Twitter.

**Meetups**

Many organizations host both in-person and online meetups for R users; you should always practice caution and be safe whenever attending meetups in-person.

* [**Local Data Analytics meetups:**](https://www.meetup.com/topics/data-analytics/) These meetups are a great way to meet other people who are interested in data analytics and build your network. These meetups are location-based, so you can connect with other data analysts in your area.
* [**R User Groups:**](https://jumpingrivers.github.io/meetingsR/r-user-groups.html)This list contains links to regional R communities, including subreddits and meetup groups. This is a useful resource if you are interested in finding R users in your area.
* [**RLadies Meetups:**](https://www.meetup.com/pro/rladies) These are in-person and virtual meetups specifically for R enthusiasts who identify as underrepresented or marginalized. These meetups are also location-based and can help you connect with other data analysts in your area.

R can be tricky to learn, but luckily there is a strong community of R users who are interested in working together and helping each other out. These resources are a good starting point if you want to begin connecting with the larger data analyst community, so take advantage of them!