

This resource is copied from the DATAQUEST website. Please use the QR code to view the website.



February 7, 2022

# Installing R on your machine

At the beginning of 2020, the amount of data in the world was estimated at [44 zettabytes](#). The amount of data generated daily is expected to reach 463 exabytes by 2025. The primary sources of these data are the following:

- **Social data** from Facebook posts, tweets, google trends
- **Machine data** from medical devices, satellites, web logs
- **Transactional data** from invoices, payment orders, payment methods, discounts

Businesses and organizations can gain a competitive advantage by analyzing large amount of data ("**big data**") to reveal patterns and gain insights. **Data analytics** studies how we collect, process, and interpret data. **Data science** applies mathematical analysis, statistical techniques, and machine learning algorithms to extract insight from data.

*Business Intelligence* (BI) applications like Power BI and Tableau help with analyzing and visualizing data. However, most of the time, data comes in unstructured format and needs preprocessing and transformation. Business Intelligence applications can't perform such transformations. Mathematical or statistical analysis cannot use them either. Powerful programming languages are necessary to perform such tasks. **R**, **Python**, and **Julia** are popular programming languages in data analytics and data science.

## What is R?

**R** is a free and open-source scripting language developed by Ross Ihaka and Robert Gentleman in 1993. It's an alternative implementation of the S programming language, which was widely used in the 1980s for statistical computing. The R environment is designed to perform complex statistical analysis and display results using many visual graphics. The [R programming language](#) is written in C, Fortran, and R itself. Most R packages are written in the R programming language, but heavy computational chunks are written in C, C++, and Fortran. R allows integration with Python, C, C++, .Net, and Fortran.

R is both a programming language and a software development environment. In other words, the name R describes both the R programming language and the R software development environment used to run R codes. R is *lexically scoped*. Lexical scoping is another name for static scoping. This means the lexical structure of the program determines the scope of a variable, not the most recently assigned variable. Here is an example of lexical scoping in R:

```
x <- 5

y <- 2

multiple <- function(y) y * x

func <- function() {

  x <- 3

  multiple(y)
```

```
}
```

```
func()
```

```
Output:
```

```
10
```

The variables `x <- 5` and `y <- 2` are global variables, and we can use them inside and outside R functions. However, when a variable is declared (`x <- 3`) inside a function (`func()`) and another function (`multiple()`) is called inside it (`func()`), the declared variable (`x <- 3`) would only be referred to if and only if the declared variable is an input in the called function (`multiple()`). It's said that the scope of the declared variable (`x <- 3`) doesn't extend into the called function (`multiple()`).

In the code snippet above, the `func()` function returns the `multiple()` function. The `multiple()` function takes *only* one input parameter, `y`; however, it uses two variables, `x` and `y`, to perform multiplication. It searches for the `x` variable globally *because* it isn't an input parameter of the function. The variable `y` is an input parameter of the `multiple()` function, so it searches for `y` locally first (called **local reasoning**). If it can't find the `y` variable declared locally, it searches globally. In the code snippet above, the `multiple()` function is evaluated with the global value of `x`, (`x <- 2`) and the global value of `y`, (`y <- 2`). It returns a value of 10.

In the next example, the `multiple()` function finds the `y` declared locally, (`y <- 10`). It uses this local value of `y` instead of the global one, (`y <- 2`). Since the variable `x` isn't an input of the `multiple()` function, it searches for its value globally (`x <- 5`). The function returns a value of 50.

```
x <- 5

y <- 2

multiple <- function(y) y * x

func <- function() {

  x <- 3

  y <- 10

  multiple(y)

}

func()

Output:

50
```

You can compile and run R on the **Windows**, **macOS X**, and **Linux** operating systems. It also comes with a *command line interface*. The **>** character represents the command line prompt. Here is a simple interaction with the R command line:

```
> a <- 5

> b <- 6

> a * b

[1] 30
```

The features of the R programming languages are organized into packages. A standard installation of the R programming languages comes with 25 of these packages. You can download additional packages from *The Comprehensive R Archive Network* ([CRAN](#)). The R project is currently being developed and supported by the R Development Core Team.

## Why use R

R is a state-of-the-art programming language for statistical computing, data analysis, and machine learning. It has been around for almost three decades with over 12,000 packages available for download on CRAN. This means that there is an R package that supports whatever type of analysis you want to perform. Here are a few reasons why you should [learn and use R](#):

- **Free and open-source:**

The R programming language is open-source and is issued under the General Public License (GNU). This means that you can use all the functionalities of R for free without any restrictions or licensing requirements. Since R is open-source, everyone is welcome to contribute to the project, and since it's freely available, bugs are easily detected and fixed by the open-source community.

- **Popularity:**

The R programming language was ranked 7th in the 2021 [IEEE Spectrum ranking](#) of top programming languages and 12th in the [TIOBE Index](#) ranking of January 2022. It's the second most popular

programming language for data science just behind Python, according to [edX](#), and it is the most popular programming language for statistical analysis. R's popularity also means that there is extensive community support on platforms like Stackoverflow. R also has a detailed online documentation that R users can consult for help.

- **High-quality visualization:**

The [R programming language is famous for high-quality visualizations](#). R's [ggplot2](#) is a detailed implementation of the grammar of graphics — a system to concisely describe the components of a graph. With R's high-quality graphics, you can easily implement intuitive and interactive graphs.

- **A language for data analytics and data science:**

The R programming language isn't a general-purpose programming language. It's a specialized programming language for statistical computing. Therefore, most of R's functions carry out vectorized operations, meaning you don't need to loop through each element. This makes running R code very fast. Distributed computing can be executed in R, whereby tasks are split among multiple processing computers to reduce execution time. R is integrated with Hadoop and Apache Spark, and it can be used to process large amount of data. R can connect to all kinds of databases, and it has packages to carry out machine learning and deep learning operations.

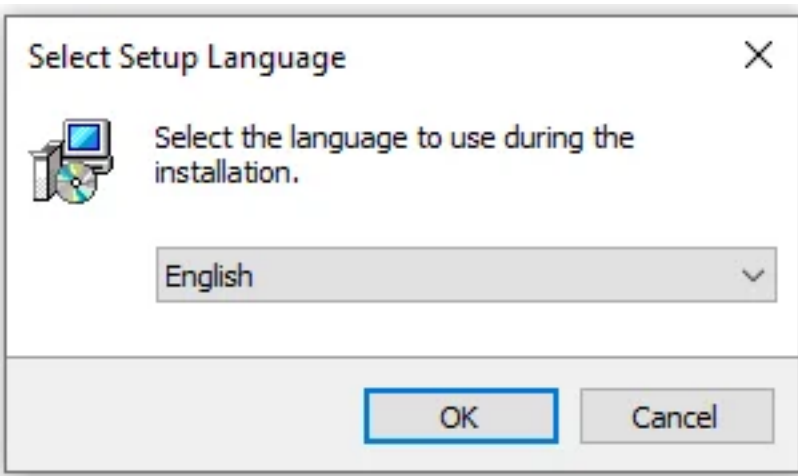
- **Opportunity to pursue an exciting career in academe and industry:**

The R programming language is trusted and extensively used in the academic community for research. R is increasingly being used by government agencies, social media, telecommunications, financial, e-commerce, manufacturing, and pharmaceutical companies. Top companies that uses R include Amazon, Google, ANZ Bank, Twitter, LinkedIn, Thomas Cook, Facebook, Accenture, Wipro, the New York Times, and many more. A good mastery of the R programming language opens all kinds of opportunities in academe and industry.

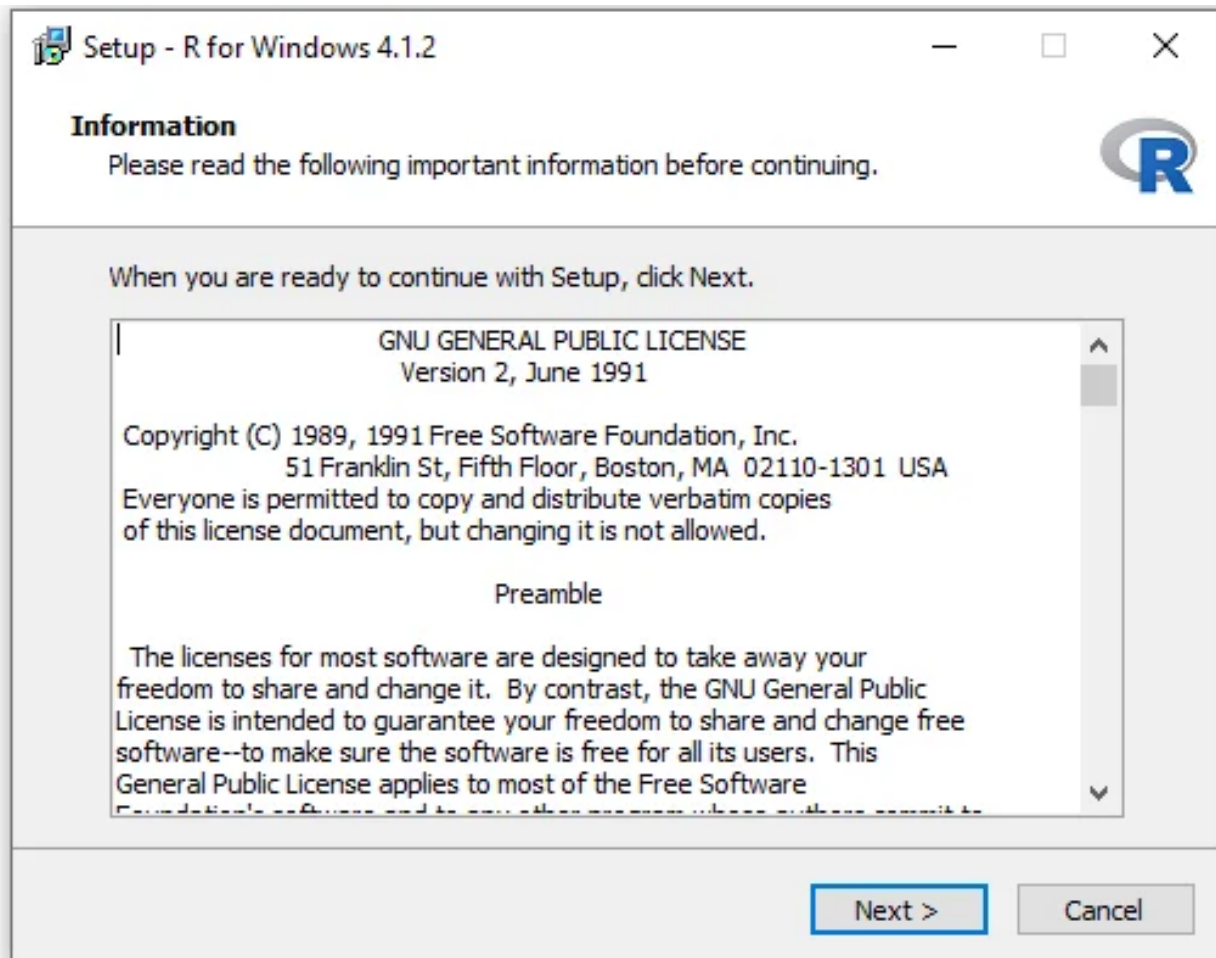
# Installing R on Windows OS

To install R on Windows OS:

1. Go to the [CRAN](#) website.
2. Click on "**Download R for Windows**".
3. Click on "**install R for the first time**" link to download the R executable (.exe) file.
4. Run the R executable file to start installation, and allow the app to make changes to your device.
5. Select the installation language.

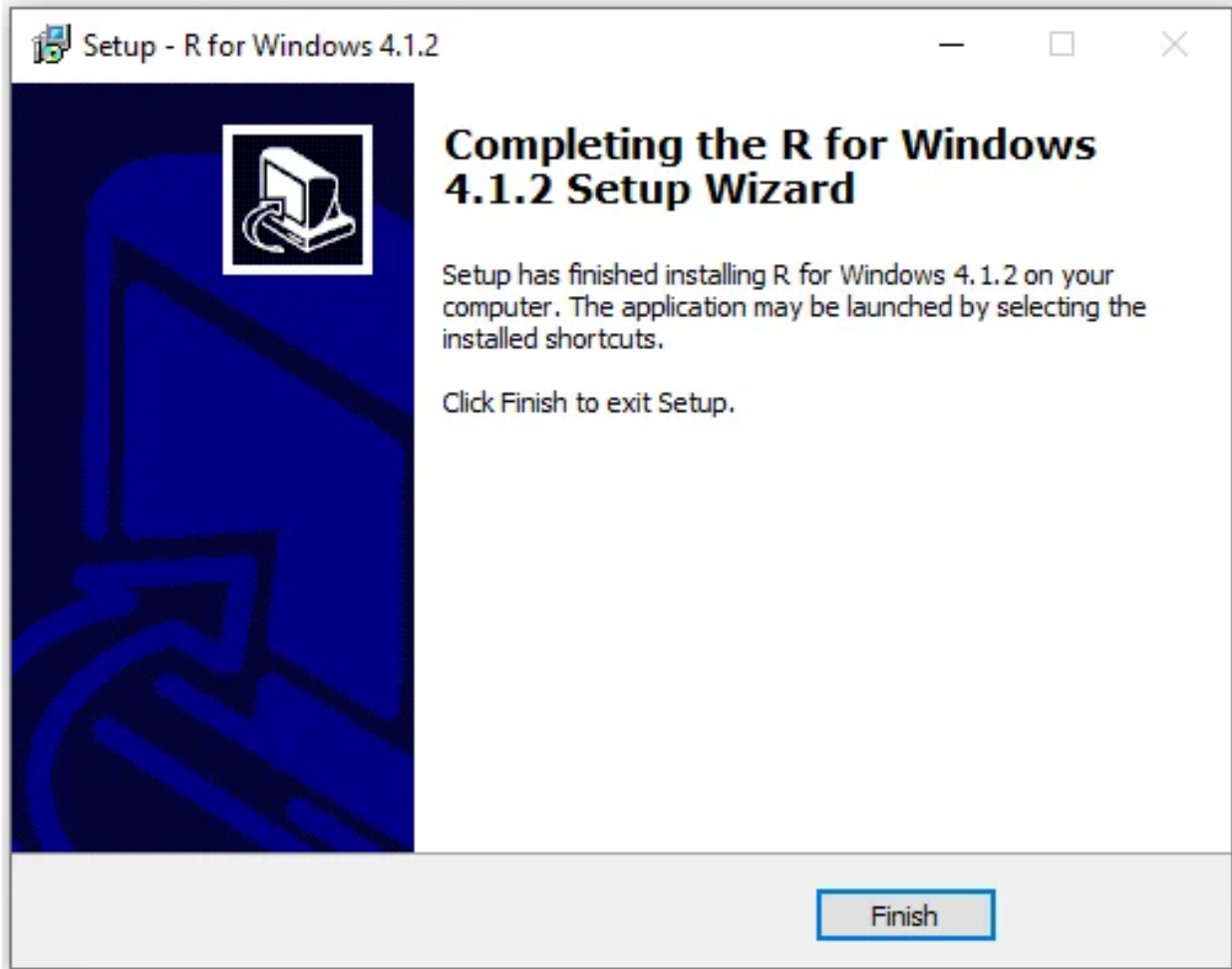


6. Follow the installation instructions.

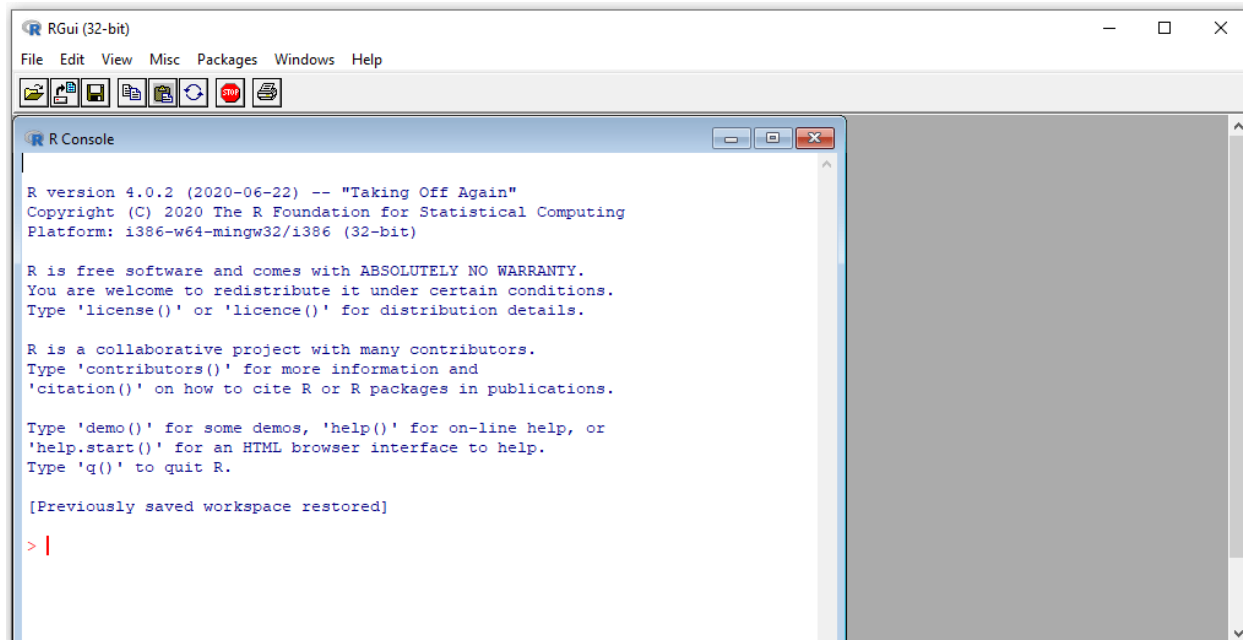


7. Click on "**Finish**" to exit the installation setup.





R has now been successfully installed on your Windows OS. Open the R GUI to start writing R codes.



## Installing R on MacOS X

Installing R on MacOS X is very similar to installing R on Window OS. The difference is the file format that you have to download. The procedure is as follows:

1. Go to the [CRAN](https://cran.r-project.org/) website.
2. Click on **"Download R for macOS"**.
3. Download the latest version of the R GUI under (.pkg file) under **"Latest release"**. You can download much older versions by following the "old directory" or "CRAN archive" links.
4. Run the .pkg file, and follow the installation instructions.

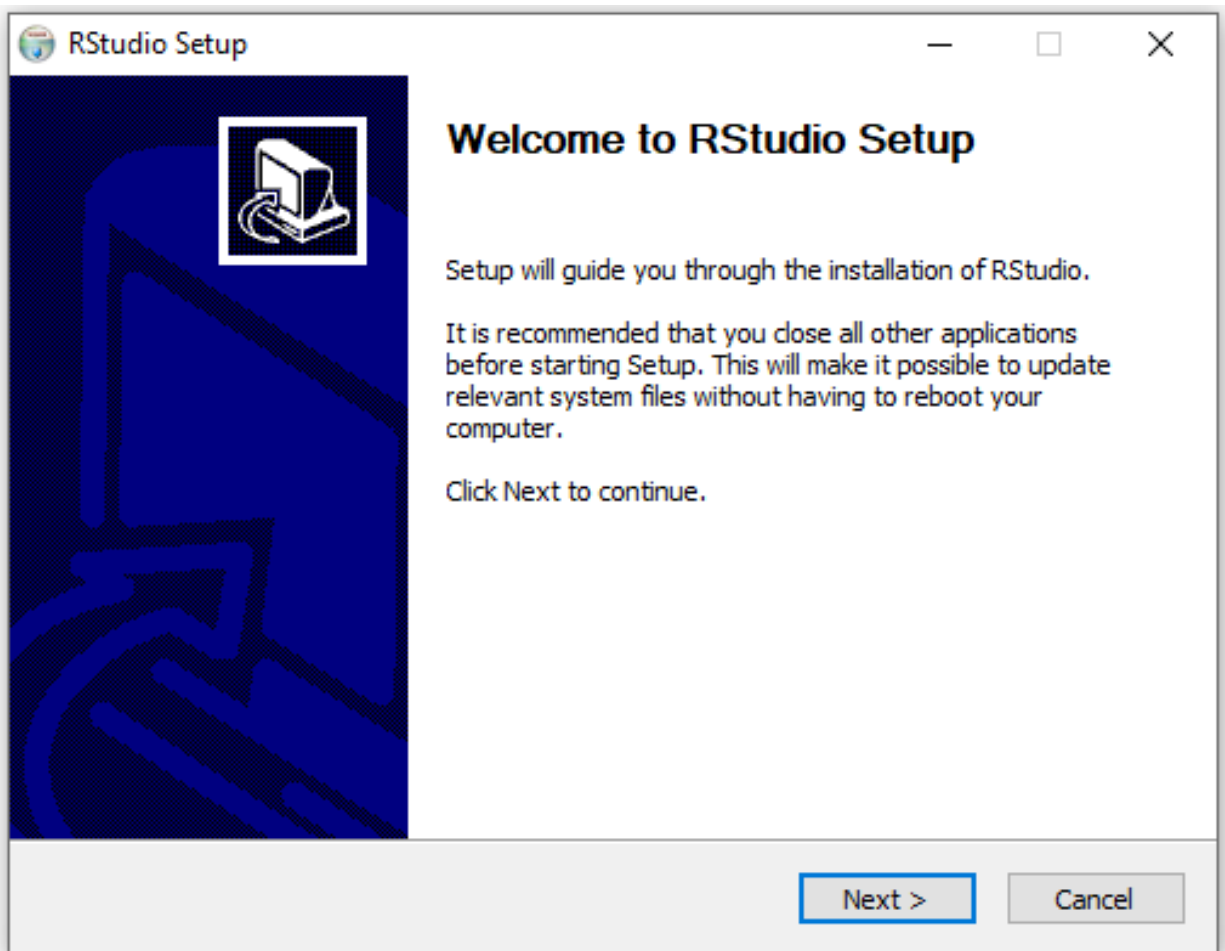
## Additional R interfaces

Other than the R GUI, the other ways to interface with R include RStudio Integrated Development Environment (**RStudio IDE**) and **Jupyter Notebook**. To run R on RStudio, you first need to install R on your computer, while to run R on Jupyter Notebook, you need to install an R kernel. RStudio and Jupyter Notebook provide an interactive and friendly graphical interface to R that greatly improves users' experience.

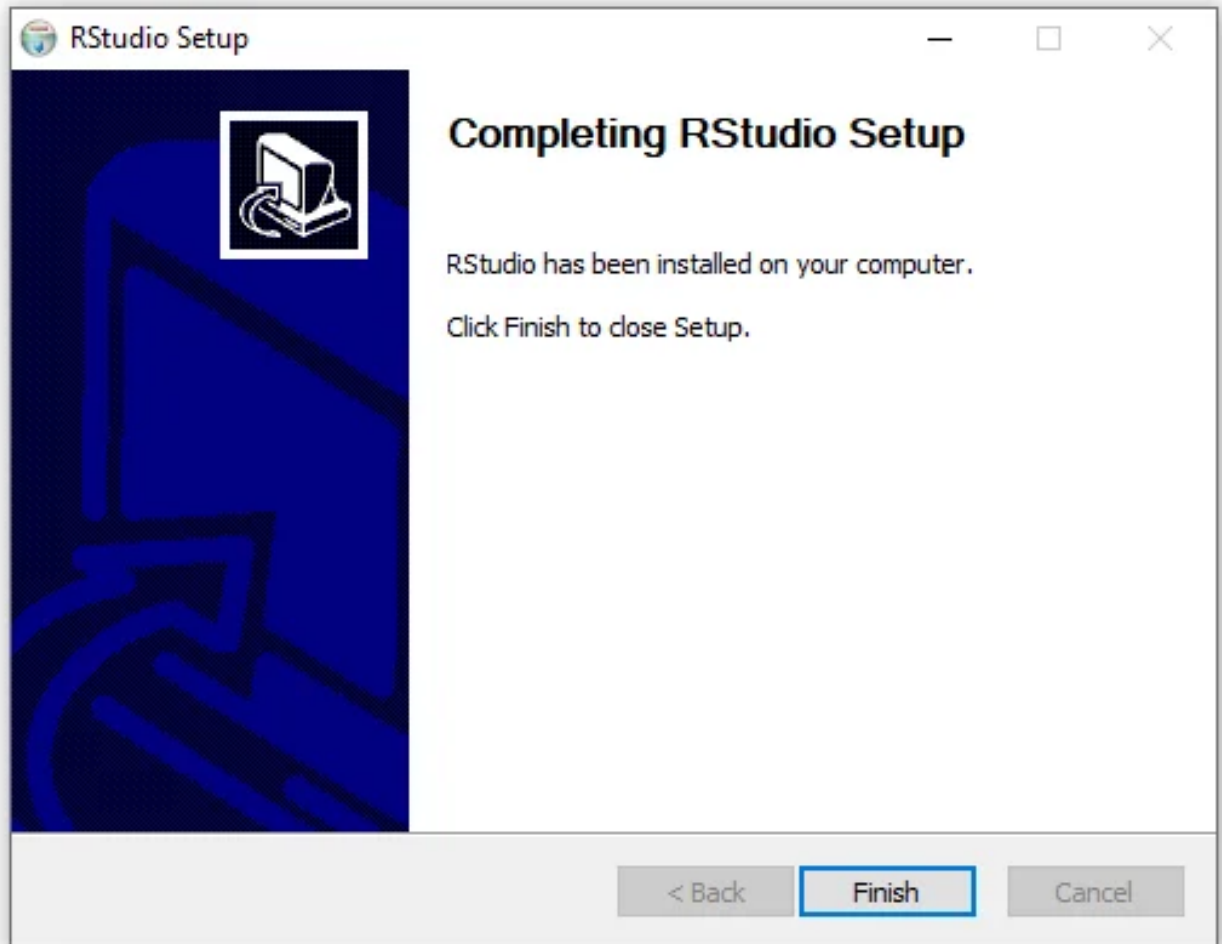
# Installing RStudio Desktop

To install RStudio Desktop on your computer, do the following:

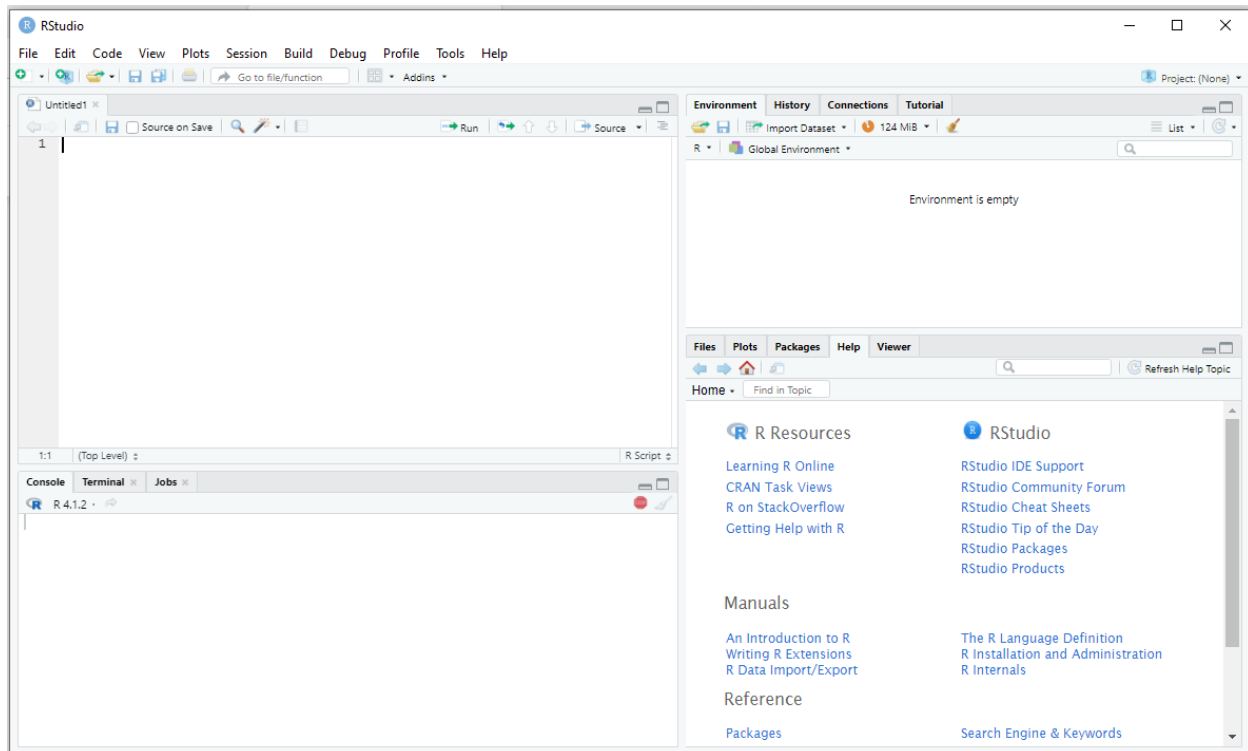
1. Go to the [RStudio](#) website.
2. Click on **"DOWNLOAD"** in the top-right corner.
3. Click on **"DOWNLOAD"** under the **"RStudio Open Source License"**.
4. Download RStudio Desktop recommended for your computer.
5. Run the RStudio Executable file (.exe) for Windows OS or the Apple Image Disk file (.dmg) for macOS X.



6. Follow the installation instructions to complete RStudio Desktop installation.



RStudio is now successfully installed on your computer. The RStudio Desktop IDE interface is shown in the figure below:



Another way to interface with R using RStudio is with the *RStudio Server*. RStudio Server provides a browser-based R interface.

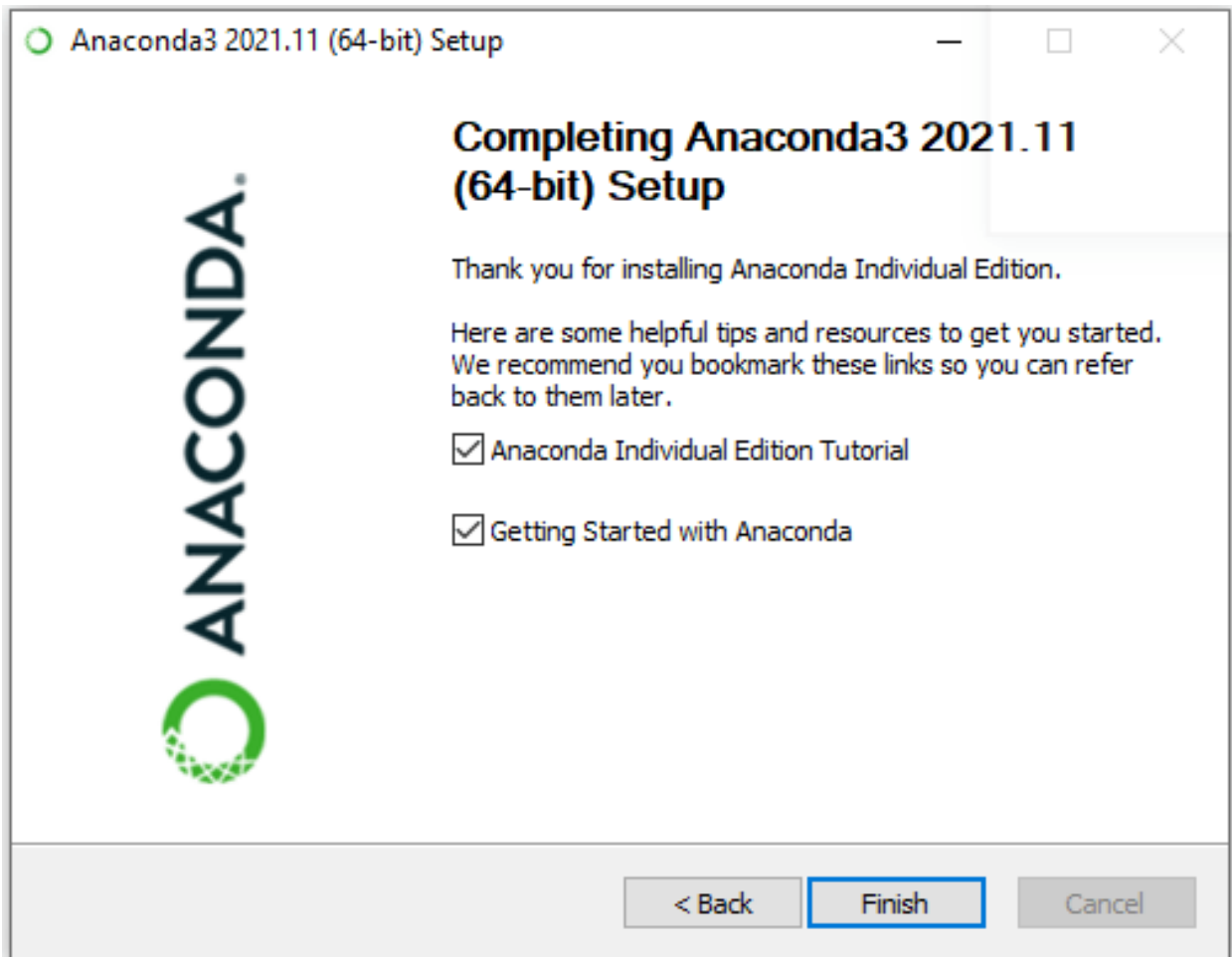
## Installing R Kernel on Jupyter Notebook for Windows OS

To install R kernel on Jupyter Notebook on your computer, do the following:

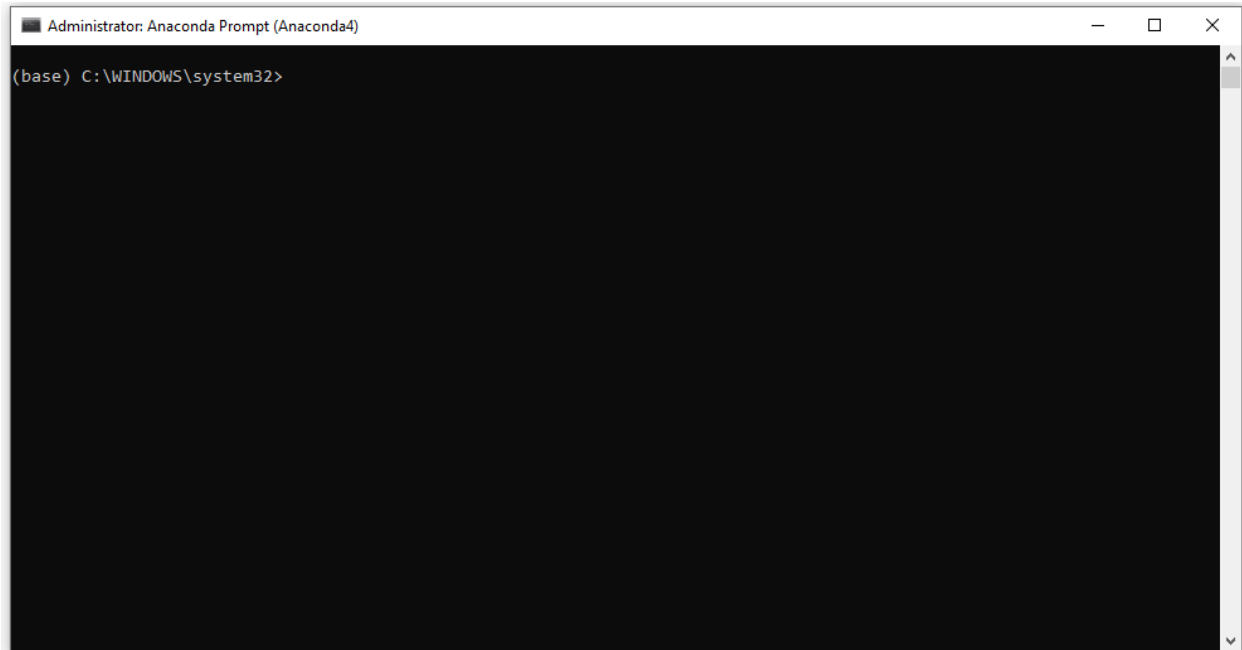
1. Download [Anaconda](#).
2. Run the downloaded file.



3. Follow the installation instructions to complete Anaconda distribution installation.



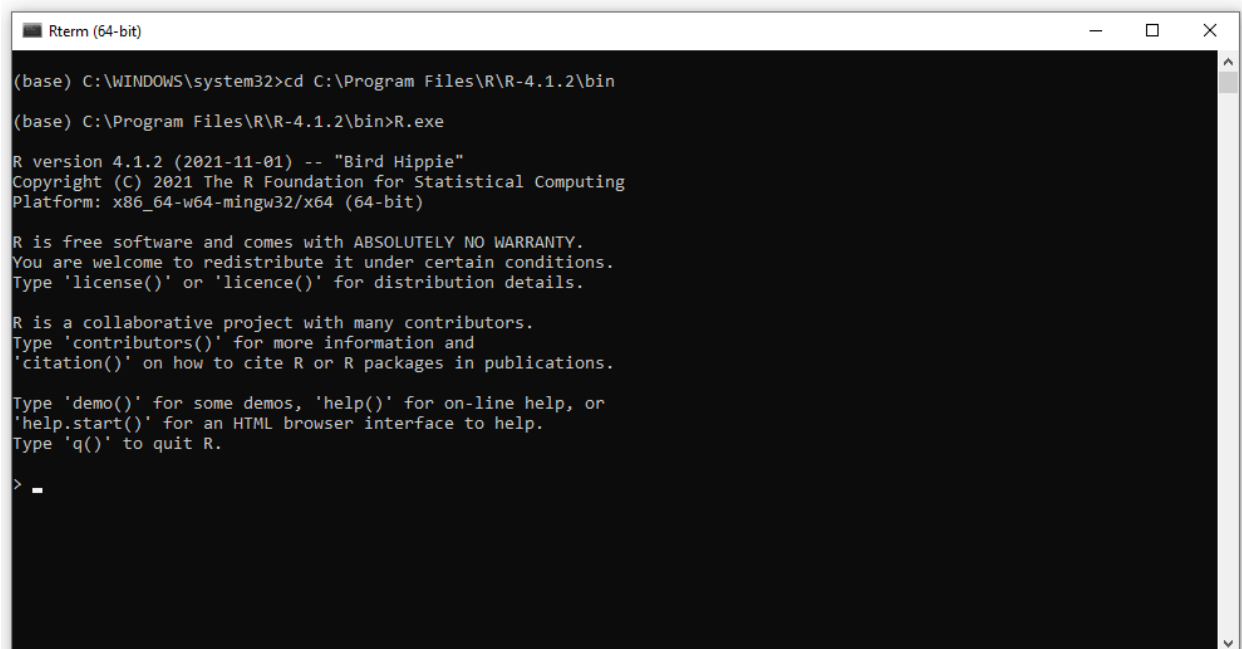
4. Open Anaconda Prompt as Administrator.



```
Administrator: Anaconda Prompt (Anaconda4)
(base) C:\WINDOWS\system32>
```

5. Change the directory to where the R.exe file is located on your computer. (The directory is `C:\Program Files\R\R-4.1.2\bin` on my computer.) Then run R from within Anaconda Prompt

6. `>cd C:\Program Files\R\R-4.1.2\bin`  
`>R.exe`



```
Rterm (64-bit)
(base) C:\WINDOWS\system32>cd C:\Program Files\R\R-4.1.2\bin
(base) C:\Program Files\R\R-4.1.2\bin>R.exe

R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> _
```



6. Install the `devtools` package with the following code to enable the `install_github()` function

```
> install.packages("devtools")
```

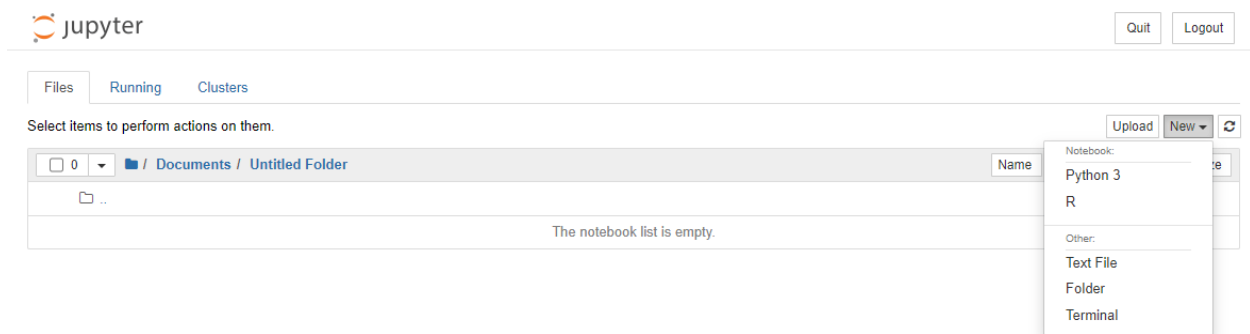
7. Install R's `IRkernel` from GitHub with the following code:

```
devtools::install_github("IRkernel/IRkernel")
```

8. Instruct Jupyter Notebook to find the `IRkernel` with the following code:

```
IRkernel::installspec()
```

9. Open Jupyter Notebook and open a New notebook with the R kernel



The steps are similar for macOS except:

- Installing the following packages along with `devtools` if they are not already installed

```
install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ',  
'devtools', 'uuid', 'digest'))
```

## Conclusion

R is an important scripting language for data analytics and data science. It's optimized for statistical analysis and outputting beautiful graphics. It's also a very fast programming language because most of its functions carry out

vectorized operations. R can be used for distributed computing to process big data, and it can be connected to different databases. You can write R codes on R GUI, Jupyter Notebook, or RStudio. Knowledge of R is important for a successful career in academia and industry.

About the author

*Dataquest*

Dataquest teaches through challenging exercises and projects instead of video lectures. It's the most effective way to learn the skills you need to build your data career.