R PROGRAMMING

FOR

BEGINNERS

DATAMENTOR

R Programming For Beginners

DataMentor

Contents

Part I

Welcome

Chapter 1

Introduction

R is an exciting language to start your programming journey. The vast applications and usage of R in todays world makes it a perfect starting point for any beginner. Being different from traditional programming languages, it feels intuitive and naturally fits into the hands of a non-programmer.

The goal of "R Programming For Beginners" is to help you as a beginner to get comfortable with R programming while learning the basics of R. By the end of this book, you'll have gained a strong footing on an emerging and powerful programming language under your belt.

1.1 What you will learn

Describing in one sentence, you'll learn to program in R. However, this book is not intended to make you a master in R or programming, but rather give you a solid foundation on the concepts and building blocks that make up R.

Depending on the system you're working on, you'll first learn to **install R** and its required tools; more specifically, the IDE: **R Studio**. You'll learn to get familiar with the working environment of R Studio and go on to write your very first program in R.

Next, you'll look into the **basics** of an R program. We'll dissect specific portions that makes up an R program and explain the working of each component. You'll learn the mechanisms of storing and retrieving your data, and perform basic logical and mathematical operations in R.

Third, you'll learn ways to store and represent different forms of data (real world and otherwise) in various **data structures** present in R, like vectors, lists, matrix, data frames and factors.

Fourth, you'll learn to change the behavior and flow of R program using **control structures**. Using control structures, you define a path for a program to take and achieve your required result.

Fifth, you'll learn about **functions**. Functions are fundamental to any process and learning to create one hands you to ability to perform tasks and maintain the structure of a program. It is undoubtely one of the most powerful tools for a programmer.

Next, you'll dig a bit deep into the **Object Oriented** (OO) world of programming, where every problem is based on the concepts of objects. You'll learn various OO features and systems R provides to simplify a complex problem in the form of objects.

Finally, you'll learn to visualize your data in the form of **Charts and Graphs**. You'll learn to create, modify and describe your data better through various plots.

1.2 Prerequisites

There are no hard prerequisites before getting started with this book. However, it's helpful if you are familiar with basic statistical concepts as we'll be talking about some through our examples. More so, we assume you to be willing to learn R.

This book doesn't assume you to be familiar with any sort of programming language before.

1.3 Things to know before start learning R

1.3.1 What is R?

R is a popular beginner friendly programming language specially built for working with data. R is built on top of the language S programming that was originally intended as a programming language that would help the student learn programming while playing around with data.

1.3.2 Why use R?

1. **R is open source and free!**

   R is free to download as it is licensed under the terms of GNU General Public license.

   You can look at the source to see what's happening under the hood. There's more, most R packages are available under the same license so you can use them, even in commercial applications without having to call your lawyer.

2. **R is popular - and increasing in popularity**

   IEEE publishes a list of the most popular programming languages each year. R was ranked 5th in 2016, up from 6th in 2015. It is a big deal for a domain-specific language like R to be more popular than a general purpose language like C#.

   This not only shows the increasing interest in R as a programming language, but also of the fields like Data Science and Machine Learning where R is commonly used.

3. **R runs on all platforms**

   You can find distributions of R for all popular platforms - Windows, Linux and Mac.

   R code that you write on one platform can easily be ported to another without any issues. Cross-platform interoperability is an important feature to have in today's computing world - even Microsoft is making its coveted .NET platform available on all platforms after realizing the benefits of technology that runs on all systems.

4. **Learning R will increase your chances of getting a job**

   According to the Data Science Salary Survey conducted by O'Reilly Media in 2014, data scientists are paid a median of $98,000 worldwide. The figure is higher in the US - around $144,000.

   Of course, knowing how to write R programs won't get you a job straight away, a data scientist has to juggle a lot of tools to do their work. Even if you are applying for a software developer position, R programming experience can make you stand out from the crowd.

5. **R is being used by the biggest tech giants**

   Adoption by tech giants is always a sign of a programming language's potential. Today's companies don't make their decisions on a whim. Every major decision has to be backed by concrete analysis of data.

6. **Companies Using R**

   R is the right mix of simplicity and power, and companies all over the world use it to make calculated decisions. Here are a few ways industry stalwarts are using R and contributing to the R ecosystem.

Table 1.1: Companies and their R contributions and applications.

Company	Application/Contribution
Twitter	Monitor user experience
Ford	Analyse social media to support design decisions for their cars
New York Times	Infographics, data journalism
Microsoft	Released Microsoft R Open, an enhanced R distribution and Microsoft R server after acquiring Revolution Analytics in 2015
Human Rights Data Analysis Group	Measure the impact of war
Google	Created the R style guide for the R user community inside Google

While using R, you can rest assured that you are standing on the shoulders of giants.

1.3.3 Is R Programming an easy language to learn?

This is a difficult question to answer. Many researchers are learning R as their first language to solve their data analysis needs.

That's the power of the R programming, it is simple enough to learn as you go. All you need is data and a clear intent to draw a conclusion based on analysis on that data.

However, programmers that come from a Python, PHP or Java background might find R quirky and confusing at first. The syntax that R uses is a bit different from other common programming languages.

While R does have all the capabilities of a programming language, you will not find yourself writing a lot of if conditions or loops while writing code in the R language. There are other programming constructs like vectors, lists, frames, data tables, matrices etc. that allow you to perform transformations on data in bulk.

1.3.4 Alternatives to learning R

R is not the only language that you can use for statistical computing and graphics. Some of the popular alternatives of R programming are:

Python – Popular general purpose language

Python is a very powerful high-level, object-oriented programming language with an easy-to-use and simple syntax.

Python is extremely popular among data scientists and researchers. Most of the packages in R have equivalent libraries in Python as well.

While R is the first choice of statisticians and mathematicians, professional programmers prefer implementing new algorithms in a programming language they already know.

The choice between R vs Python also depends on what you are trying to accomplish with your code. If you are trying to analyze a dataset and present the findings in a research paper, then R is probably a better choice. But if you are writing a data analysis program that runs in a distributed system and interacts with lots of other components, it would be preferable to work with Python.

SAS (Statistical Analysis System)

SAS is a powerful software that has been the first choice of private enterprise for their analytics needs for a long time. Its GUI and comprehensive documentation, coupled with reliable technical support make it a very good tool for companies.

While R is the undisputed champion in academics and research, SAS is extremely popular in commercial analytics. But R and Python are gaining momentum in the enterprise space and companies are also trying to move towards open-source technologies. Time will tell if SAS will continue its dominance or R/Python will take over.

SPSS – Software package for statistical analysis

SPSS is another popular statistical tool. It is used most commonly in the social sciences and is considered the easiest to learn among enterprise statistical tools.

SPSS is loved by non-statisticians because it is similar to excel so those who are already familiar with it will find SPSS very easy to use.

SPSS has the same downside as SAS – it is expensive. SPSS was acquired by IBM in 2009 for a reported $1.2 billion.

1.4 Getting Help

Learn and get help from others. There are tons of great R communities that will help you solve real life problems and become better in R.

Some of them are:

StackOverflow Most Popular programming Q&A site on the web

R-bloggers R news and tutorials contributed by (750) R bloggers

R-help An active mailing list for R

Nabble R specific mailing list and forum

Chapter 2

Run R on Your Computer

You will find the easiest way to run R programming on your system (Windows, Mac OS X or linux) in this chapter.

2.1 In Mac

1. Go to official site of R programming
2. Click on the CRAN link on the left sidebar
3. Select a mirror
4. Click "Download R for (Mac) OS X"
5. Download the latest pkg binary
6. Run the file and follow the steps in the instructions to install R.

2.2 In Linux

On Ubuntu

The Advanced Packaging Tool (APT) that comes with Ubuntu uses a file called sources.list to decide where to search for packages.

Before we can install R, we need to tell Ubuntu to look into the CRAN R repositories and also add a public key for secure download.

1. Open the sources.list file (usually located at /etc/apt/sources.list) in a text editor and add the following line at the end

   ```
   deb https://<my.favorite.cran.mirror>/bin/linux/ubuntu <distribution>/
   ```

   For instance, If you are running Ubuntu trusty and want to use the RStudio CRAN mirror, the line would be

   ```
   deb https://cran.rstudio.com/bin/linux/ubuntu trusty/
   ```

   If you are lazy like all good programmers, you can do this directly from the terminal without having to open a text editor as

   ```
   sudo sh -c 'echo "deb http://cran.rstudio.com/bin/linux/ubuntu trusty/" >> /etc/apt/sources.list'
   ```

2. **Authenticate the Ubuntu packages on CRAN**

   The packages for Ubuntu that are stored on CRAN mirrors are all signed using a key with ID E084DAB9 We download the public key from the Ubuntu keyserver using this ID and add it to our system using the command

   ```
   sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E084DAB9
   ```

3. **Update the list of available packages**

   Since we modified the sources.list, we need to tell APT to download the packages that are available from the CRAN servers by running the command.

   ```
   sudo apt-get update
   ```

4. **Download and install R**

   Almost done. Just download and install the R package by running the command:

   ```
   sudo apt-get -y install r-base
   ```

5. Open up the R console and issue following command.

   ```
   $ R
   ```

   If there were no issues during installation. The R console should open successfully with information about your R installation.

RedHat-based Distributions

The process is similar for Redhat-based Linux Distributions like CentOS. Instead of modifying a file like sources.list, you can directly add the repository for EPEL(Extra Packages for Enterprise Linux) by using the following command.

```
su -c 'rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm'
```

You can find the url for the correct rpm file for your system here.

Now it's just a matter of updating the list of available packages and installing R.

```
sudo yum update
sudo yum install R
```

Fedora

Installing R on fedora is a piece of cake. The Fedora repositories have the latest version of R binaries installed.

Just run the commands:

```
sudo yum update
sudo yum install R
```

2.3 In Windows

1. Go to official site of R programming
2. Click on the CRAN link on the left sidebar
3. Select a mirror
4. Click "Download R for Windows"

5. Click on the link that downloads the base distribution
6. Run the file and follow the steps in the instructions to install R.

2.3.1 Should I Install 32 bit or 64 bit?

Most people don't need to worry about this. Obviously the 64-bit version of R won't work on a 32-bit machine but both the 32-bit and 64-bit versions of R runs seamlessly on 64-bit Windows.

You might want to consider installing 32-bit version of R if your production environment is 32-bit because some packages might have compatibility issues and might cause the "But it works on my machine" fiasco.

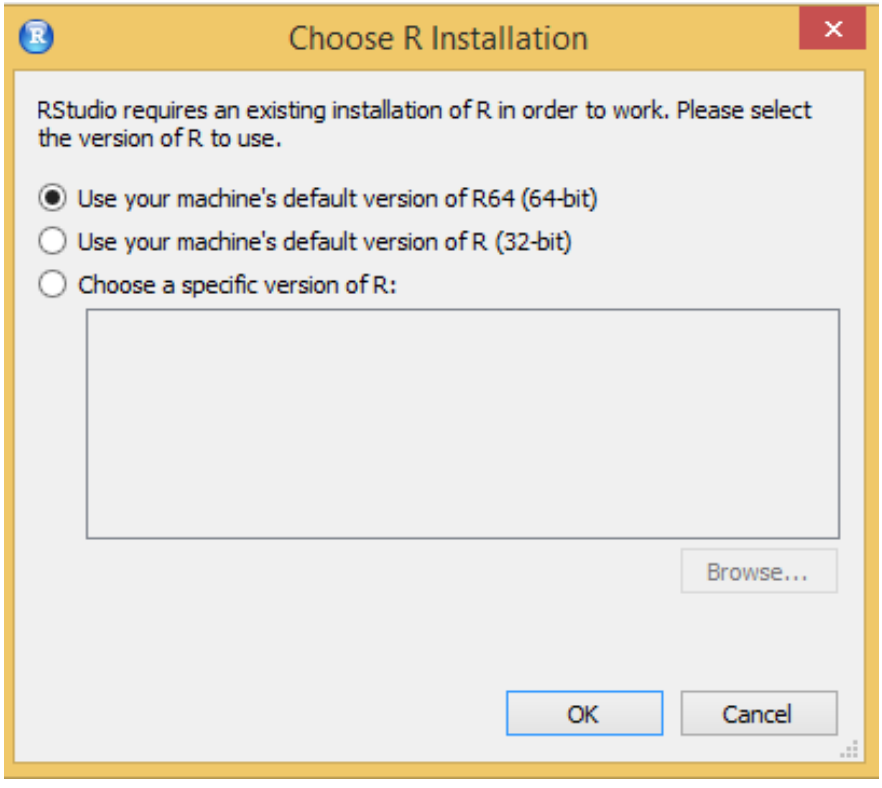
2.4 Installing RStudio

RStudio is the most popular IDE for running R programs and has a free license.

Download the RStudio (Windows, Linux and Mac OS X), run the file and follow the instructions to install it.

Note: R should be installed on your system before you can run RStudio.

After you install RStudio and open it for the first time, it will ask you to choose which version of R to use.

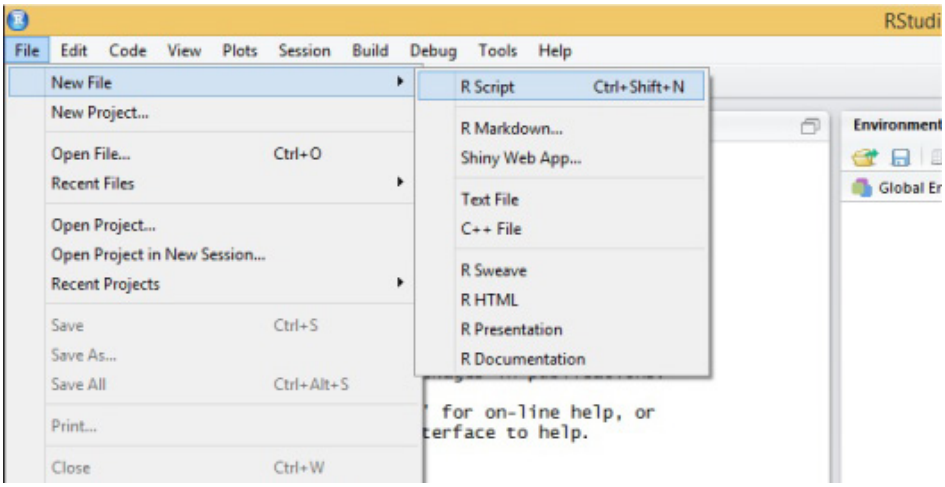


If RStudio detects that R hasn't been installed on your system, it will show you a warning.

If R has been installed, you'll see the R Studio interface. In the beginning, you can only see the R console where you can write one line statements in R and execute them.

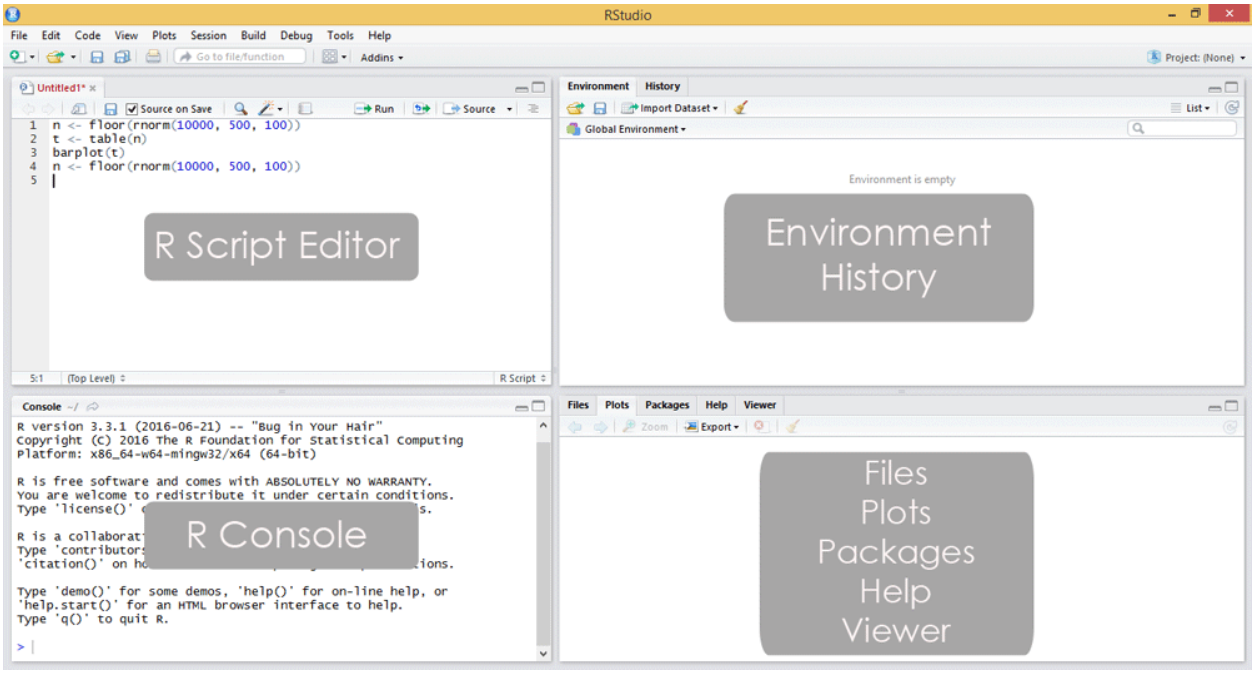
However, even for trivial work, you will need to perform a sequence of steps and it is better to create an R script.

Go to **File > New File > R Script** as shown in the screenshot below to create a new R script.



You can now see the R Script Editor where you can type and save R programs that span multiple lines. RStudio isn't just a text editor but an IDE that helps you run and debug R scripts with ease.

The R Studio GUI is divided into 4 major sections as shown in the screenshot below:



Part II

Basics

Introduction

Now that you have a basic idea of what is R, why choose it over others, it's time to dig deep into it.

We'll start off with the very basics of R. The following chapters proceeds as follows:

- In Your first R program, as the name suggests, you'll learn to write and execute your first R program. You'll witness the power of R with just a few lines of code.
- In Variables and constants, you'll learn different ways to store and retrieve your valuable data in R. You'll learn what are variables and constants, how they work, different kinds of data you can store, and reserved words you cannot use as variables and constants.
- In Output, you'll learn to print variables and constants in R. You'll learn to print variables using variable name, `print()` function and `cat()` function.
- In Operators, you'll learn tools to perform basic operations in R, like addition, subtraction, comparisions, and assignments.
- In Comments, you'll learn to leave comments about your program. This will be an integral part of your programming which helps to better describe and document the working of your program.
- In Packages, you'll learn to access tools that enhances the power of R. You'll learn about both local and remote packages, and how to download them, import them and finally use them in R.
- In Environment, you'll learn about environments in R. You'll see how variables and constants are stored in R.

Chapter 3

Your First R Program

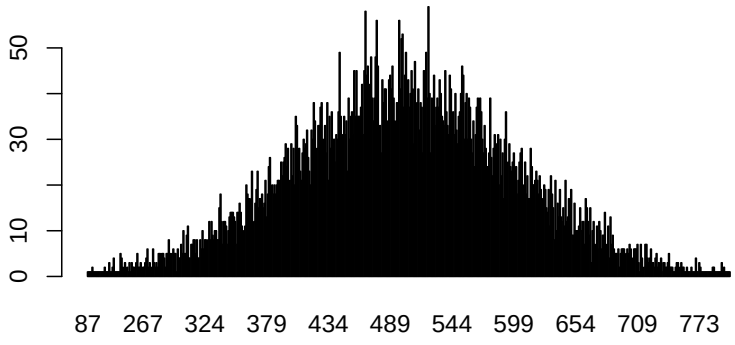
R holds a reputation for getting things done with very little code. If you're a programmer and thinking "Here comes the Hello World code", you're in for a surprise.

In just three lines of code, your first R program will generate 10,000 numbers in a random distribution, organize them based on frequency and create a fancy barchart.

Copy the following code into the RStudio window, press **Ctrl+A(Windows)** or **Cmd+A(Mac)** to select all three lines and press **Ctrl+Enter(Windows)** or **Cmd+Enter(Mac)**. This wil run the given R program.

```
n <- floor(rnorm(10000, 500, 100))
t <- table(n)
barplot(t)
```

Look at the right bottom section of RStudio and you will see this beautiful bar graph showing the bell curve of a random normal distribution.



Here's what each part of the code does:

Chapter 42

3D Plot

There are many functions in R programming for creating 3D plots. In this section, we will discuss on the `persp()` function which can be used to create 3D surfaces in perspective view.

This function mainly takes in three variables, `x`, `y` and `z` where `x` and `y` are vectors (Chapters 11) defining the location along x- and y-axis. The height of the surface (z-axis) will be in the matrix `z`. As an example,

Let's plot a cone. A simple right circular cone can be obtained with the following function.

```
cone <- function(x, y){
 sqrt(x^2+y^2)
}
```

Now let's prepare our variables.

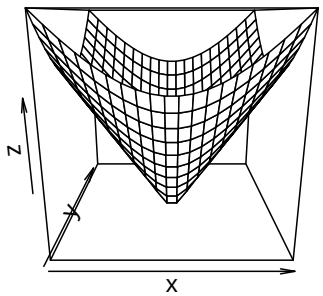
```
x <- y <- seq(-1, 1, length= 20)
z <- outer(x, y, cone)
```

We used the function `seq()` to generate vector of equally spaced numbers.

Then, we used the `outer()` function to apply the function cone at every combination of `x` and `y`.

Finally, plot the 3D surface as follows.

```
persp(x, y, z)
```



Adding Titles and Labeling Axes to Plot

We can add a title to our plot with the parameter `main`.

Similarly, `xlab`, `ylab` and `zlab` can be used to label the three axes.

Rotational angles

We can define the viewing direction using parameters `theta` and `phi`.

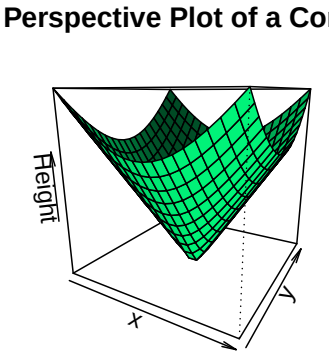
By default `theta`, azimuthal direction, is 0 and `phi`, colatitude direction, is 15.

Coloring and Shading Plot

Coloring of the plot is done with parameter `col`.

Similarly, we can add shading with the parameter `shade`.

```
persp(x, y, z,
  main="Perspective Plot of a Cone",
  zlab = "Height",
  theta = 30, phi = 15,
  col = "springgreen", shade = 0.5)
```



Part IX

Conclusion

Chapter 43

What Next

Depending on your professional background and your personal interests, there are many paths that you can take from here.

Data Science

One of the heavy uses of R is in the field of Data Science. Almost every company bases their decisions on their data. Whether a business plans to add a new store, or a hospital analyzes the health condition of a patient, meaningful data is key to each decision. And, drawing meaningful insights from data is Data Science.

With the advent of IoT devices creating terabytes and terabytes of data that can be used to make better decisions, data science is a field that has no other way to go but up. Infact, Harvard Business Review named data scientist the "sexiest job of the 21st century".

We've created a course on Getting Started in Data Science with R to give you hands-on knowledge on Data Science. The course will give you a solid foundation to explore this amazing field. More specifically we answer,

- What is Data?
- What is Data Science?
- What learn Data Science?
- And most importantly, how to get started in the field?

And as you start to learn the process, you'll find the use of R very intuitive and natural to the process.

Statistical computing

R is the most popular programming language among statisticians. In fact, it was initially built by statisticians for statisticians. It has a rich package repository with more than 9100 packages with every statistical function you can imagine.

R's expressive syntax allows researchers – even those from non computer science backgrounds to quickly import, clean and analyze data from various data sources.

R also has charting capabilities, which means you can plot your data and create interesting visualizations from any dataset.

Machine Learning

One of another paths that is trending all around the world is Machine Learning. In simple words, machine learning is giving machine the ability to learn on its own. It is done through various algorithms that parses data; the machine then learns from the data and take informed decisions on its own.

Companies like Amazon, Google, Facebook's inner systems is driven by machine learning algorithms to the extent that it's integrated and reflected through their core values.

R has found a lot of use in predictive analytics and machine learning. It has various package for common ML tasks like linear and non-linear regression, decision trees, linear and non-linear classification and many more.

If you're fascinated by machine learning, you can start exploring machine learning algorithms and applications online.

Advanced R

Whichever path you choose, learning to program is inherent to each of them and writing effective code gives you a much needed headstart to solve new problems quickly. So, it's only logical to invest your next step in learning advanced topics in R through various courses and tutorials.