Fundamentals of Mathematics and Statistics with R

Dr. Priyanga D. Talagala

2020-07-09

Contents

Chapter 1

Introduction

1.1 Installing R

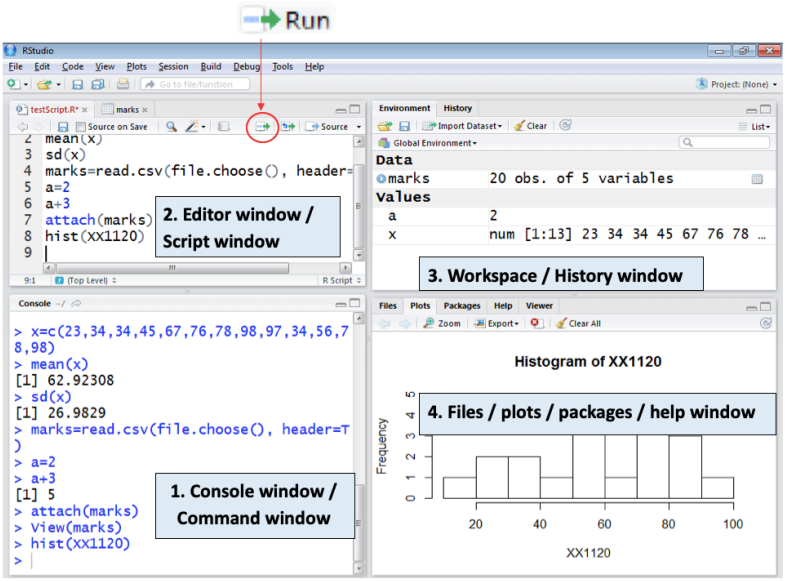
- **Step 1:** First download R freely from the Comprehensive R Archive Network (CRAN) https://cran.r-project.org/. (At the moment of writing, R 4.0.2 is the latest version. Choose the most recent one.)
- **Step 2:** Then install R Studio's IDE (stands for integrated development environment), a powerful user interface for R from https://rstudio.com/products/rstudio/download/. Get the Open Source Edition of RStudio Desktop. RStudio allows you to run R in a more user-friendly environment.
  - You need to install **both** R and Rstudio to use RStudio.
  - If you have a pre-existing installation of R and/or RStudio, I highly recommend that you re install both and get as current as possible.
- **Step 3:** Then open **Rstudio**.

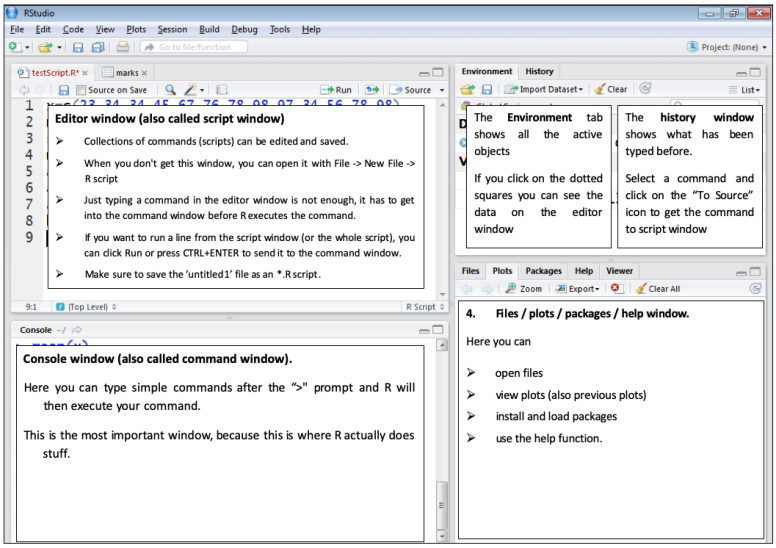
1.2 RStudio layout

The RStudio interface consists of four windows (see Figure 1 and 2).

1. Bottom left: console window (also called command window). **This is where you type and run all your R commands**
2. Top left: editor window (also called script window).
3. Top right: workspace / history window.

4. Bottom right: Files / plots / packages / help window.





Now you are familiar with the layout. Let's begin with R basics.

1.3 Installing an R Package

- The primary location for obtaining R packages is CRAN
- Packages can be installed with the `install.packages()` function in R
- To install a single package, pass the name of the package to the `install.packages()` function as the first argument

The following the code installs the `tidyverse` package from CRAN

```
install.packages("tidyverse")
```

- This command downloads the `tidyverse` package from CRAN and installs it on your computer
- Any packages on which this package depends will also be downloaded and installed
- **Installing the tidyverse package could take several minutes. You only need to do this once**.

1.4 Loading an R Packages

- Installing a package does not make it immediately available to you in R; you must load the package
- The `library()` function is used to load packages into R
- The following code is used to load the tidyverse package into R
- **NOTE:** Do not put the package name in quotes!

```
library(tidyverse)
```

- Some packages produce messages when they are loaded (but some don't)

Chapter 2

Differentiation

First, we take the equation as an expression

```
f <- expression(x^2)
```

To calculate first derivative of f, we use `D()` function and `x` to specify that derivation has to be carried out with respect to x.

```
f_1 <- D(f, "x")
print(f_1)
```

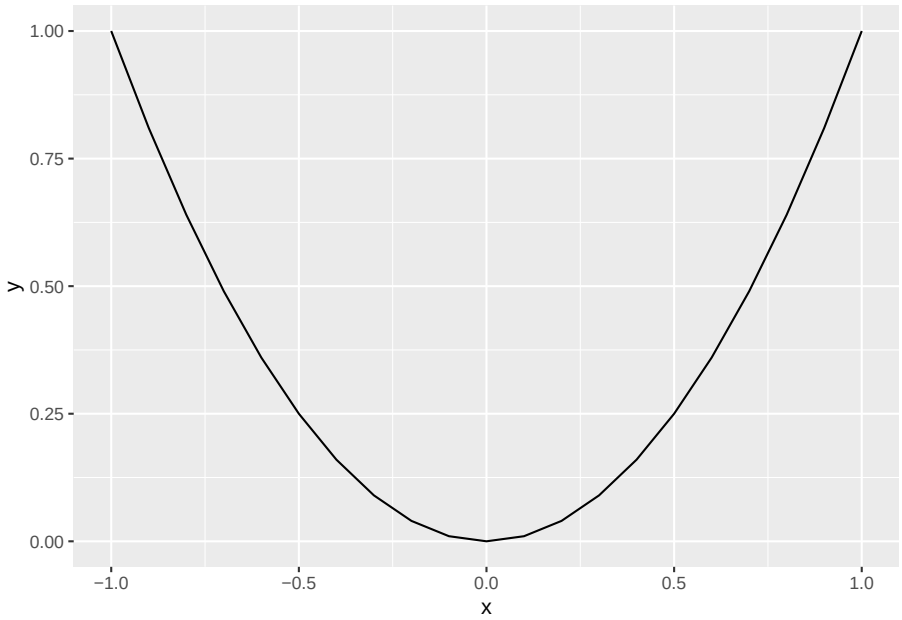
```
## 2 * x
```

Sketch the graph of f and f'

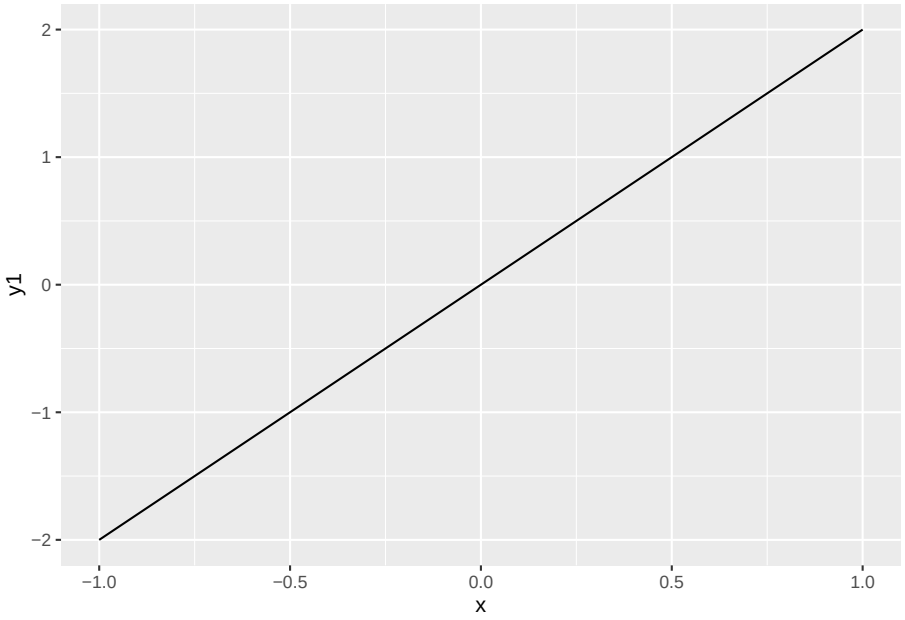
```
library(ggplot2)
x <- seq(-1, 1, by = 0.1)
y <- eval(f)
x <- seq(-1, 1, by = 0.1)
y1 <- eval(f_1)
data <- data.frame(x, y, y1)
head(data)
```

```
##      x    y   y1
## 1 -1.0 1.00 -2.0
## 2 -0.9 0.81 -1.8
## 3 -0.8 0.64 -1.6
## 4 -0.7 0.49 -1.4
## 5 -0.6 0.36 -1.2
## 6 -0.5 0.25 -1.0
```

```
p <- ggplot(data, aes(x = x, y = y)) +
  geom_line()
print(p)
```




```
q <- ggplot(data, aes(x = x, y = y1)) +
  geom_line()
print(q)
```



2.1 Higher Derivatives

The following R command can be used to find second derivative of the above f.

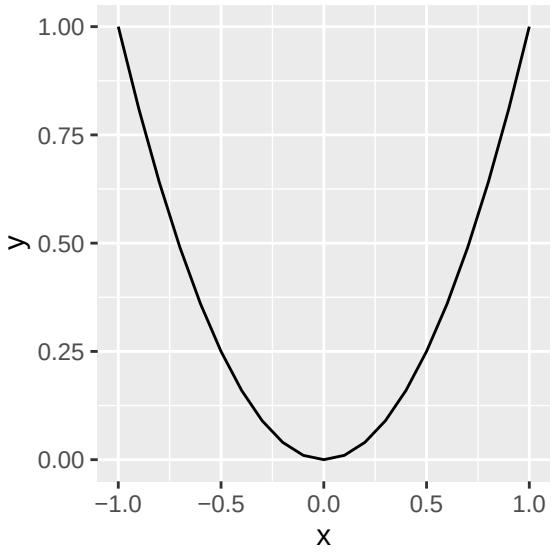
```
f_2 <- D(D(f, "x"), "x")
print(f_2)
```

```
## [1] 2
```

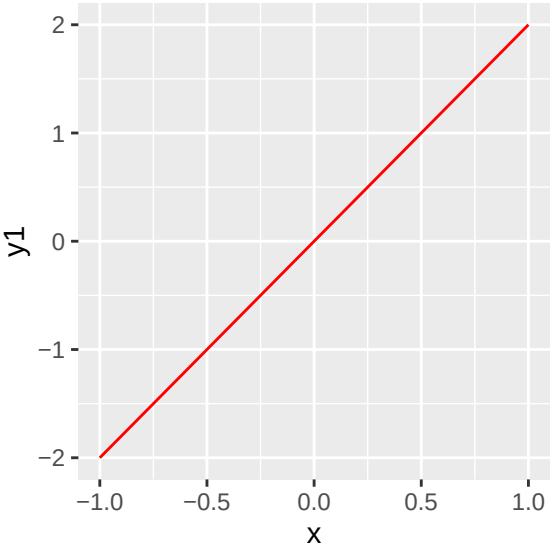
```
x <- seq(-1, 1, by = 0.1)
y2 <- eval(f_2)

data <- data.frame(x, y, y1, y2)
head(data)
```

```
##      x    y   y1 y2
## 1 -1.0 1.00 -2.0  2
## 2 -0.9 0.81 -1.8  2
## 3 -0.8 0.64 -1.6  2
## 4 -0.7 0.49 -1.4  2
## 5 -0.6 0.36 -1.2  2
## 6 -0.5 0.25 -1.0  2
```

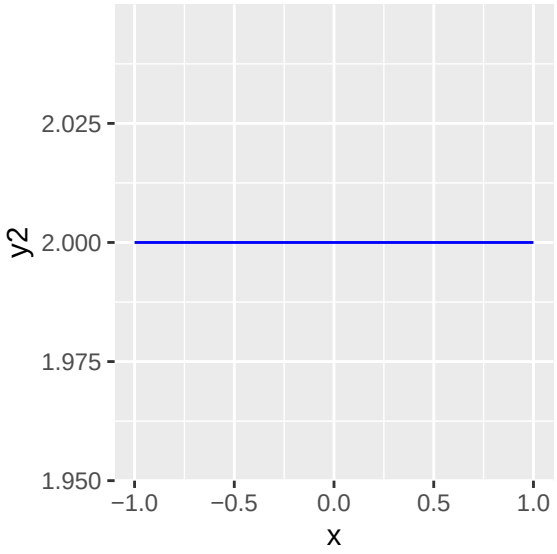
```
p <- ggplot(data, aes(x = x, y = y)) +
  geom_line()
print(p)
```




```
q <- ggplot(data, aes(x = x, y = y1)) +
  geom_line(colour = "red")
print(q)
```



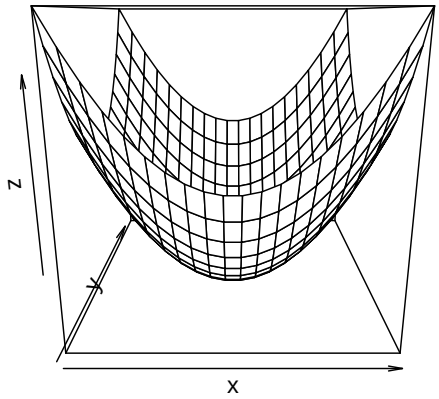
```
r <- ggplot(data, aes(x = x, y = y2)) +
  geom_line(colour = "blue")
print(r)
```



2.2 Partial Derivatives

If the expression is having more than one independent variable, we can calculate differentiation with respect to each of them.

```
f <- expression(x^2 + y^2)

x <- y <- seq(-3, 3, length = 20)
surface <- function(x, y) {
  eval(f)
}
z <- outer(x, y, surface)
persp(x, y, z)
```



Differentiate with respect to `x`

```
D(f, "x")
```

```
## 2 * x
```

Differentiate with respect to `y`

```
D(f, "y")
```

```
## 2 * y
```

Chapter 3

Statistical Distributions

Chapter 4

Applications

Chapter 5

Final Words

We have finished a nice book.