Chapter 1 Introduction to R

1.1 Getting Started

The simplest way to use R is to use it as if it were a calculator. For example, if we want to know what one plus one is, you may type:

```
1 + 1
```

We can use any arithmetic operator, like addition, subtraction, multiplication, divison, exponentiation, and modulus operations:

```
# addition
1 + 1

# subtraction
6 - 4

# multiplication
2 * 2

# division
10 / 5
10.2 / 5

# integer divison
15.2 %/% 5
15.7 %/% 5

# modulus
15.2 %% 5
15.8 %% 5

# exponential
2^3
```

R also provides numerous built-in functions to use in calculations, such as natural logs, exponentiation, square root, absolute value:

```
# natural log
log(10)

# exponentiation
exp(2)

# square root
sqrt(4)

# absolute value
abs(-4)
```

R includes extensive facilities for accessing documentation and searching for help. This is useful to get more information about a specific function. The `help()` function and `?` help operator in R provide access to the documentation pages for R functions. For example, to get help with the round() function, we submit the following code:

```
#help() function
help(round)

#? operator
?round()
```

1.2 Variable Assignment

A basic concept in programming is called a variable. A variable allows you to store a value (e.g. 10) or an object (e.g. a function description) in R. We can then further use the variable's name to access the value or the object that is stored within this variable.

For example, you can assign a value 10 to a variable `my_variable`:

```
my_variable <- 10
```

To print out the value of the variable, you simply type the name of your variable:

```
my_variable
```

A valid variable name consists of letters, numbers and the dot or underline characters. The variable name starts with a letter or the dot not followed by a number:

```
# A variable can be operated on
my_variable + 1

# And passed to a function
sqrt(my_variable + 1)

# To reassign a variable, just reassign in
my_variable <- 3000

# You can also operate on and reassign a variable to itself
my_variable <- my_variable + 1
```

You can broaden assignments beyond numbers:

```
result <- sqrt(9)

fruit_1 <- "apple"
fruit_2 <- "banana"
fruit_3 <- "cantaloupe"
```

1.3 Finding Variables

To know all the variables currently available in the workspace we use the `ls()` function:

```
print(ls())
```

The `ls()` function can also use patterns to match the variable names.

```
# List the variables starting with the pattern "var".
print(ls(pattern = "var"))
```

1.4 Deleting Variables

Variables can be deleted by using the `rm()` function. Below we delete the variable my_variable:

```
my_variable <- 5
rm(my_variable)
```

All the variables can be deleted by using the `rm()` and `ls()` function together:

```
rm(list = ls())
```

Another common way to remove all variables in the R environment is to click on the little broom icon next to the button `Import Dataset` under the Environment tab. One can also go up to `Session` and do `Restart R` or `New Session` (the `Restart R` and `New session` options might come in handy for when programs crash).