Workshop 1: An Introduction to R

Myles D. Garvey, Ph.D

Winter, 2020

1 Learning Objectives

1. To leverage basic mathematical operations in R
2. To apply the use of variables in R
3. To apply the use of functions in R
4. To leverage data structures in R
5. To learn logical control in R

2 Workshop Description and Submission

In this workshop, we will use R to solve a variety of example problems so as to illustrate the fundamentals of R. Our course will leverage a lot of R, and hence, a foundational knowledge is absolutely necessary. This workshop will comprise of five mini-tutorials to help guide you through the basics. Completion of the workshop is part of your grade, so please ensure that you have installed R and that you are ready to write commands!

You will submit to blackboard a .PDF document. Complete each exercise in this workshop. These are easy and small, and so it should not take you very long. In a word document, type up the question number, the original question text, and your response in R code. Save your word document as a .PDF and submit it to the submission link on Blackboard. Please note that failure to submit this as a .PDF will result in a 0 on the workshop.

3 Tutorial 1: Basic Math in R

In this tutorial, you will learn how to use R as a calculator, as well as how to write mathematical expressions using numbers and variables. To motivate this usage, let us consider a simple mathematical problem. We all know that the basic equation for the profit of a firm for a particular product that it sells is equal to the price per unit times the total number of units sold less the cost per unit times the total number of units produced. Suppose we sell a product for $5 per unit,

and that we sell 100 of these. Furthermore, suppose that we make a product for $3 per unit, and that we manufacture 110 of these. What is our total profit? We can use R as a calculator to find our answer.

```
> 5*100-3*110
[1] 170
```

Listing 1: Using R as a Calculator to find total firm profit.

We can use the R console as a calculator by multiplying (*), dividing (/), subtracting (-), adding (+), taking the exponent (ˆ), and finding the remainder of a number after dividing by another number (%%). For example, suppose we want to take into account inflation and find the present value of the profit if we were to make the same amount but 10 years from now. This happens by compounding roughly 98% of our profit for 10 years. That is, our $170 in 10 years from now is worth $170(0.98)^{10}$. To find this in R, we have:

```
>170*(0.98)^10
[1] 138.9024
```

Listing 2: Present value of our profit in 10 years from now.

However, we can generalize our math by using a *variable*. Recall that a variable is anything that stores a number, letter, string, or other information. Variables are useful, since we can assign a value to them ahead of time, and not have to worry about changing their values in equations later on. In our problem, we can define a few variables such as price and cost per unit, as well as the number of units produced and the number of units sold, as well as the total profit, revenue, and cost. We can accomplish this in R by using either the assignment operators (=) or the more frequent assignment operator (<-):

```
> price<-5
> cost<-3
>
> units_sold<-100
> units_produced<-110
>
> TR<-price*units_sold
> TC<-cost*units_produced
>
> profit<-TR-TC
>
> interest_rate<-0.02
> compounding_factor<-1-interest_rate
> num_of_years<-10
>
> profit_years<-profit*compounding_factor^num_of_years
> profit
[1] 170
> profit_years
[1] 138.9024
```

Listing 3: Using variables to do our math on profits.

In conclusion, we can see that a variety of mathematical operations can be used in R just as how we use a calculator. We are able to also use variables to simplify our analysis down, and more easily change them. Now try to replicate this, but on your own:

Exercises

1. Repeat the same computations, but this time with the price set to \$8, cost set to \$3, units sold set to 150, and the units produced set to 150. Also set the number of years to 5.

4 Tutorial 2: Using Functions in R

In this tutorial, you will learn how to use functions in R. In the last section, you learned how to use R as a calculator by using basic mathematical operations as well as variables. However, this is not the only use of R. In R, we can use a function to carry out a sequence of commands to manipulate a given input into a new output. In R, and in programming generally, we often take what is called a *black-box approach*. That is, we do not care *how* the steps are performed when we use a function. The only thing we care about is what the function does, what inputs are required, and what type of output can be expected.

In our problem above, we can use a function to change our interest rate. Usually, we do not use 0.02, but rather something that is called a continuously compounded factor. This factor is usually e^{-r}, where r is the interest rate. Notice that this is the typical exponential function from algebra. We can call this function in R by using it's function name, and subsequently providing the input. In our instance, the input would be the minus of our interest rate, while the output would be the compounding factor. Doing this, we obtain:

```
> price<-5
> cost<-3
>
> units_sold<-100
> units_produced<-110
>
> TR<-price*units_sold
> TC<-cost*units_produced
>
> profit<-TR-TC
>
> interest_rate<-0.02
> compounding_factor<-exp(-1*interest_rate)
> num_of_years<-10
>
> profit_years<-profit*compounding_factor^num_of_years
> profit
[1] 170
> profit_years
[1] 138.9024
>compounding_factor
[1] 0.9801987
```

Listing 4: Using variables to do our math on profits with the exponential function.

Exercises

2. In the code above, you can rewrite the term for profit_years by simply removing the exponent and putting it inside the exponential function. That is, you can first find the factor by setting it equal to e^{-rt}, where r is the interest rate and t is the number of years, and then you can find the profit years by multiplying the profit times the factor. Carry out these steps using the exponential function with the input adjusted accordingly.

5 Tutorial 3: Using Logical Control in R

In this tutorial, you will learn how to use logical control such as for loops. Suppose we would like to find the profit from earlier, but over 10 years, each year compounded to take into account the devaluation of the dollar. That is, assume we make the same profit every year. However, since the value of a dollar always changes (and more specifically decreases over time), we need to take this devaluation into account each year. One way to do this is to store each of these into their own variable and repeat the command over and over again:

```
> price<-5
> cost<-3
>
> units_sold<-100
> units_produced<-110
>
> TR<-price*units_sold
> TC<-cost*units_produced
>
> profit<-TR-TC
>
> interest_rate<-0.02
> compounding_factor<-1-interest_rate

> num_of_years<-1
> profit_years_1<-profit*compounding_factor^num_of_years
> num_of_years<-2
> profit_years_2<-profit*compounding_factor^num_of_years
> num_of_years<-3
> profit_years_3<-profit*compounding_factor^num_of_years
> num_of_years<-4
> profit_years_4<-profit*compounding_factor^num_of_years
> num_of_years<-5
> profit_years_5<-profit*compounding_factor^num_of_years
> num_of_years<-6
> profit_years_6<-profit*compounding_factor^num_of_years
> num_of_years<-7
> profit_years_7<-profit*compounding_factor^num_of_years
> num_of_years<-8
> profit_years_8<-profit*compounding_factor^num_of_years
> num_of_years<-9
> profit_years_9<-profit*compounding_factor^num_of_years
> num_of_years<-10
> profit_years_10<-profit*compounding_factor^num_of_years
> profit_years_1
```

```
[1] 166.6
> profit_years_2
[1] 163.268
> profit_years_3
[1] 160.0026
> profit_years_4
[1] 156.8026
> profit_years_5
[1] 153.6665
> profit_years_6
[1] 150.5932
> profit_years_7
[1] 147.5813
> profit_years_8
[1] 144.6297
> profit_years_9
[1] 141.7371
> profit_years_10
[1] 138.9024
```

Listing 5: Math on profits for 10 years with devaluation.

This obviously becomes too cumbersome. Hence, we can use a for loop to simplify our math, as well as more succinctly organize our data. We do this by using the `for` command, with the parentheses. Inside the parentheses, we put the logical control statement. In for loops, we do this by defining an index variable for which the loop will iterate and assign to values in a given vector (more on vectors in the next section). Hence, we can simplify the previous code by doing the following:

```
> price<-5
> cost<-3
>
> units_sold<-100
> units_produced<-110
>
> TR<-price*units_sold
> TC<-cost*units_produced
>
> profit<-TR-TC
>
> interest_rate<-0.02
> compounding_factor<-1-interest_rate

> for(i in 1:10){
+     profit_years<-profit*compounding_factor^i
+     print(profit_years)
+ }
[1] 166.6338
[1] 163.3342
[1] 160.1
[1] 156.9298
[1] 153.8224
[1] 150.7765
[1] 147.7909
```

```
[1] 144.8644
[1] 141.9959
[1] 139.1842
```

Listing 6: Math on profits for 10 years with devaluation using a for loop.

Exercises

3. Use the for loop to print the profit values over the years 1 to 5 with the information provided from the previous exercises.

6 Tutorial 4: Using Vectors in R

In this tutorial, you will learn how to use vectors in R to be able to store more information in variables than just single values. A vector is like a mailbox system. We define vectors using the c function. The input into this function is the data for which we would like to store. Each datapoint is given an *index*, which allows us to refer back to the data point by using a combination of the name we have given to the vector as well as its position in the vector (i.e. the index). In our problem, we can replace the for loop from earlier by using a simple vector and vector math:

```
> price<-5
> cost<-3
>
> units_sold<-100
> units_produced<-110
>
> TR<-price*units_sold
> TC<-cost*units_produced
>
> profit<-TR-TC
>
> interest_rate<-0.02
> compounding_factor<-1-interest_rate

> num_of_years<-c(1,2,3,4,5,6,7,8,9,10)
> profit_years<-profit*compounding_factor^num_of_years
> profit_years
 [1] 166.6000 163.2680 160.0026 156.8026 153.6665 150.5932
 [7] 147.5813 144.6297 141.7371 138.9024
>profit_years[8]
[1] 144.6297
```

Listing 7: Math on profits for 10 years with devaluation but using vectors to do so.

Notice that we first define the vector by using the c function. This function allows us to insert data into a vector (check this for yourself!). We then use the vector in our math. What R will do is apply the equation that uses our vector to each and every element in the vector. It will then return a new vector with the result in each corresponding index. Notice that we last checked to see what the value of the profit would be in 8 years from now. This is stored in the 8th position of the vector. We were able to determine what is in this position of the vector by using the bracket notation, and subsequently providing the index inside the brackets.

Exercises

4. Repeat the same computations using the vector, but this time with the price set to \$8, cost set to \$3, units sold set to 150, and the units produced set to 150. Do this with years 1 to 5 instead of the 1 to 10 like we did earlier.

5. Swap out the `c(1,2,3,4,5)` by using the `seq` function. If you do not know what this function does, check the help guide in R (or, Google it!). Ensure to have the correct inputs into this function.

7 Tutorial 5: Defining Functions in R

The last tutorial will show you how to define functions in R. You recently came across how to use functions in R from an earlier tutorial. However, there are occasions where writing your own functions will help save you time. This is because function writing will allow you to organize and generalize your code. It also allows you to reuse code without having to copy and paste all over the place. First, we want to be able to define a function that will compute the total net present value over a given amount of time when provided the per-unit profit, cost, total units produced, total units sold. We will also allow the user to provide an input for the interest rate.

The way in which we define functions in R is as follows. First, you need to give your function a name, much how you do with variables. Once you have decided on a name for your function, you set it equal to the function function (no, that's not a mistake. The function `function` takes as inputs the inputs you intend to have for your function. If you want an input to be an *optional input*, that is, an input that the user can choose to not provide, just simply set the input value to a default value. The function `function` then is followed by a block of code, which uses the variables provided as the inputs, that R will execute when your function is *called*. For example, we can define the profit function as follows:

```
> total_profit = function(price,cost,sold,produced){
+     TR<- price*sold
+     TC<- cost*produced
+     profit<-TR-TC
+     return(profit)
+ }
> total_profit(5,3,100,110)
[1] 170
```

Listing 8: A profit function.

Notice how in this block of code, we define the function `total_profit`, which takes 4 inputs that the user of the function must provide: price, cost, sold, produced. Since none of these variables are set equal to an explicit default value, it is assumed that all of these must absolutely be provided by the user. Notice the function `return` which takes a value as input. This function will provide it's input as the output value back to the user whom called your function. Notice how the function is used in the very last line. We simply use its name, along with the inputs of 5,3,100,110. We will now use the function to define a new function which will compute the net present value over time:

```
> net_present_value = function(price,cost,sold,produced,interest_rate = 0.02,num_
    of_years=10){
+     years<-1:num_of_years
+     compounding_factor<- exp(-interest_rate*years)
+     profit_years<-profit(price,cost,sold,produced)*compounding_factor
+     profit_years<- sum(profit_years)
+     return(profit_years)
+ }
>
> net_present_value(5,3,100,110)
[1] 1525.432
> net_present_value(5,3,100,110,0.01,5)
[1] 824.9612
```

Listing 9: Math on profits for 10 years with devaluation but using vectors to do so.

Notice that we define the function with 4 required inputs and 2 optional inputs. If the user does not specify the optional inputs, they are automatically assigned to 0.02 and 10, respectively. Once the user provides these inputs, R will assign the values to each of the variable names provided in the function definition. The first line in the function generates a vector of values from 1 to the number of years under consideration. The second line finds the compounding factors for each year t, which is defined as e^{-rt}, as we discussed earlier. The third line uses our new profit function to compute the profit, which is then normalized to take into account the devaluation by multiplying by the compounding factor. Last, we add together all of the profits, since each profit in each year is now in terms of today's money. This sum is then returned. We can see that now the net present value is very easily computed using our function definition.

6. Use this new function to compute the net present value of the information provided in the previous exercises.

7. Create a new function that will use the net present value function to compute the average discounted profit per year over a given number of years.