Lab 2 Notes

Jon Ashbrock

January 24, 2018

1 Introduction

After today's class students will be able to

1. Use a while-loop in **R**
2. Use the else-if statement in **R**
3. Pick random variables from a discrete distribution
4. Begin to understand the concept of probabilistic modeling

2 The While-loop

The while-loop is the other main type of loop that exists in almost every programming language. The other type of loop we learned, the for-loop, is especially useful when you know exactly how many times you want the loop to repeat. However, we don't always a priori know how many times the loop will repeat. The while loop will run arbitrarily many times, so long as some condition is satisfied. The condition is any true-or-false value that you specify. The syntax is as follows:

```
while (BOOLEAN VALUE){
# Statements to be repeated
}
```

Recall that a Boolean Value is variable which is either true or false. The loop runs so long as the value inside of the parameter is true. If you write the following code in **R**, the loop will run for forever

```
while (TRUE){
print('Hello World')
}
```

We will be using the while-loop in this lab homework assignment in which you simulate a baseball game. The while loop in the homework will be exactly the command we need to allow as many hitters as possible while there are fewer than 3 outs.

Discussion 1. *Write a program that checks whether or not a number eventually reaches* 1 *in the Collatz algorithm. Print how many iterations it took.*

3 If-else Statements

We have already taught the use of the if-statement in a previous lab. The If-else statement does exactly what it says: it does one thing if the boolean value is true and another if it is false. Here is the syntax:

```
if (Boolean Value){
Condition to do if Boolean is true
} else {
Condition to do if Boolean was false
}
```

You can also use the "else if" statement to check more than 2 things at once. Suppose I want to write a program that looks at a number, prints negative if it is negative, zero if it is zero, and positive if it is positive. The following code will do that.

```
Number <- 100
if (Number <0){
print("Negative")
} else if (Number ==0){
print("zero")
} else {
print("Positive")
}
```

4 Random Variable from Discrete Distributions

So far we have seen how to generate random numbers uniformly from an interval and how to select random integers uniformly from a range of numbers. Now we turn our attention to picking random objects from a finite number of categories, each with a given probabibility of being picked. We can do this with the uniform distribution on $[0, 1]$ and with $if - else$ statement. As a simple example, the following code will simulate flipping a coin and outputting the result:

```
random <- runif(1,0,1)
if(random <= 0.5){
print("heads")
} else {
print("tails")
}
```

The above algorithm works because the odds of the variable "random" being less than or equal to 0.5 is 50%. If we wanted to simulate flipping a coin that was weighted to be 60% heads, we could chance the above code to check if the random number is less than 0.6. The only other case we need to consider is if there are more than two options. Suppose we have a coin that has a 40% chance of being heads or tails and a 20% chance of landing on its side. We can simulate this with the following code:

```
random <-runif(1,0,1)
if(random <= 0.4){
print('heads') }
else if (random <= 0.6){
print('side') }
else {
print('tails')}
```

Discussion 2. *Why does this do what we want it to do?*

5 Probabilistic Modeling

Often times in industry, we will ask questions where the answer is impossible or unfeasible to calculate. Examples of such questions include

1. How many extra votes will our candidate receive because of this advertising campaign?
2. How much money will our company save by hiring this worker via increased productivity?
3. How many extra wins will our baseball team get if each of our players increases their batting average by 2%

These questions are almost impossible to calculate but we can use *probabilistic models* to estimate what the answer is. Because of the nature of probability, if we set up the appropriate experiment to model the above situations and run the experiment hundreds of times, our answer will likely converge to the answer we are trying to calculate. The third problem is appropriate difficulty for us to model at this point at our current level of **R** knowledge. It is the subject of our next few homework assignments.