Introduction to Programming: the R perspective

Peter D Smits

April 24, 2013

What is programming?

Structure of Interpretation of Computer Programs by Abelson and Sussman 1996 page 1.

> *We are about the study the idea of a computational process. Computational processes are abstract beings that inhabit computers. As they evolve, processes maniputate other acstract things called data. The evoution of a process is directed by a pattern of rules called a program. People create programs to direct processes. In effect, we conjure the spirits of the computer with our spells.*

Continued. . .

What is programming?

A computation process is indeed much like a sorcerer's idea of a spirit. It cannot be seen or touched. It is not composed of matter at all. However, it is very real. It can perform intellectual work. It can answer questions. It can affect the world by disbursing money at a bank or by controlling a robot arm in a factory. The programs we use to conjure processes are like a sorcerer's spells. They are carefully composed from symbolic epressions in arcane and esoteric programming languages that prescribe the tasks we want our process to perform.

R: a brief history

Console and scripts

Using the console or REPL

Writing our first script

Flow control

Writing our first function

Using what we know now, let's write our own useful function.

Let's duplicate the sum function.

Writing our first function

First, let's find out what sum does.

```
x <- seq(5)
sum(x)

## [1] 15


y <- c(1, 1)
sum(y)

## [1] 2
```

Writing our first function

```r
sum.prime <- function(x) {
  y <- 0  # create variable we can increase in value

  # use loop through every value in x and add it to y
  for(i in seq(length(x))) {
    # seq is for making sequences
    # length determines how long a vector is

    y <- y + x[i]
  }

  # return determines the output of a function
  return(y)
}
```