What is Programming?

A program is a list of instructions for the computer to follow.

(Just like a recipe is a list of instructions for the chef to follow.)

What is Programming?

Two steps to writing a program:

1. Figure out in English exactly what you want the computer to do
2. Translate line by line to computer code

First Example

Pseudocode:

Task: Given a list of numbers L, return the mean.

Steps:

```
1 Set s = 0
2
3 for each number N in L:
4   Set s = s + N
5
6 Set mean = s / length(L)
```

But the computer is stupid! What if L is a list with 0 numbers?

First Example

Better Pseudocode:

Task: Given a list of numbers L, return the mean.

Steps:

```
1  if length(L) = 0:
2    Set mean = 0
3
4  else:
5    Set s = 0
6
7    for each number N in L:
8      Set s = s + N
9
10   Set mean = s / length(L)
```

Flow control: if/else, for, while

If Statements

Flow Control:

If statement pseudocode:

```
if (condition):
  [do this]
else:
  [do that]
```

In R:

```
if (condition) {
  [do this]
} else {
  [do that]
}
```

For Loops

Flow Control:

For statement pseudocode:

```
for every element x in a set S:
  [do something]
```

In R:

```
for (x in s) {
  [do something]
}
```

While Loops

Flow Control:

For statement pseudocode:

```
1 while (condition stays true):
2   [do something]
```

In R:

```
1 while (condition) {
2   [do something]
3 }
```

Example in R

Task: Compute the mean of a list.

```
1  L = c(1,84,5,17,5,28)
2
3  if (length(L) == 0) {
4          mean = 0
5  } else {
6          s = 0
7          for (number in L) {
8                  s = s + number
9          }
10         mean = s / length(L)
11 }
```

Note: "`a = [something]`" sets the value of the variable a. In R, people often use "`a <- [something]`". These are equivalent.

Conditionals

Example of conditionals for if statements and while loops:

- `a < 5`
- `a <= 5`
- `a > 5`
- `a >= 5`
- `a == 5` (NOT `a = 5`)
- `a != 5` (means "a $\neq$ 5")

Vectors (lists) and Comments

R works principally with what it calls *vectors*, but you can think of these as lists. To make a list of numbers, use the c command: ("c" for combine)

Putting a # in front of a line makes it a comment. It has no effect on the code, but it helps you remember what each part does.

```
1 # creates a list with three elements
2 mylist = c(10,5,4)
3
4 print(mylist)
5
6 # adds the number 100 to the list
7 mylist = c(mylist, 100)
8
9 print(mylist)
```

Functions

A function is a self-contained bit of code that takes in some values as input, and returns some output.
Task: Function for the mean of a list.

```
1  findthemean = function(L) {
2          if (length(L) == 0) {
3                  return(0)
4          } else {
5                  s = 0
6                  for (number in L) {
7                          s = s + number
8                  }
9                  return(s / length(L))
10         }
11 }
```

Debugging

It's impossible to program perfectly the first time!
Task: Compute the mean of a list.

```
1  findthemean = function(L) {
2          if (length(L) == 0) {
3                  return(0)
4          } else {
5                  s = 0
6                  for (number in L) {
7                          s = number
8                  }
9                  return(s / length(L))
10         }
11 }
```

Debugging

It's impossible to program perfectly the first time!
Task: Compute the mean of a list.

```
1  findthemean = function(L) {
2          if (length(L) == 0) {
3                  return(0)
4          } else {
5                  s = 0
6                  for (number in L) {
7                          s = number
8                          print("Sum so far is:")
9                          print(s)
10                 }
11                 return(s / length(L))
12         }
13 }
14
15 # see also cat, paste, paste0
```

Another Function Example

```
1  whichisbigger = function(x,y,z) {
2    if (x > y && x > z) {
3            return(x)
4    } else {
5            if (y > z) {
6                    return(y)
7            } else {
8                    return(z)
9            }
10   }
11 }
```

Note: "&&" means "AND", "||" means "OR", "!" means "NOT"

Editing in R Studio

You can type one command at a time, and hit enter to see the result.

If you need to type in multiple lines before executing (like for a function), use Shift+enter.

When writing anything longer, use the editor window of R Studio. To run, click Code $\rightarrow$ Source.

Other important functions

- `sample` – picks a number randomly from a list, generate a random permutation of a list
- `runif` – pick a random real number between 0 and 1
- `is.element(element, list)` – returns True if element is in list, False otherwise

Most important tool: Google.
Reputable answers: stackoverflow.com

Other resources

- See course page for other handouts
- "Try R" free course on codeschool.com
- Me!