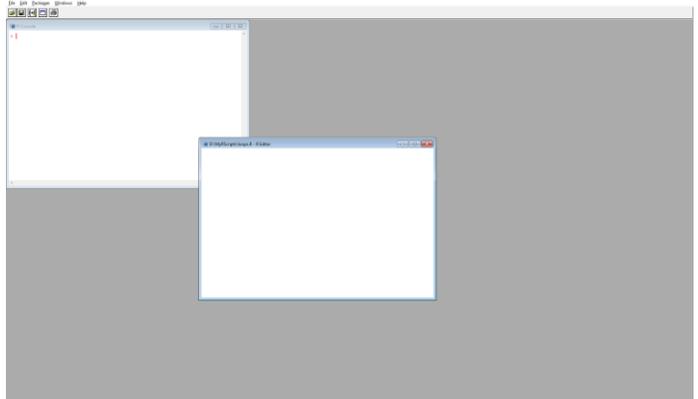
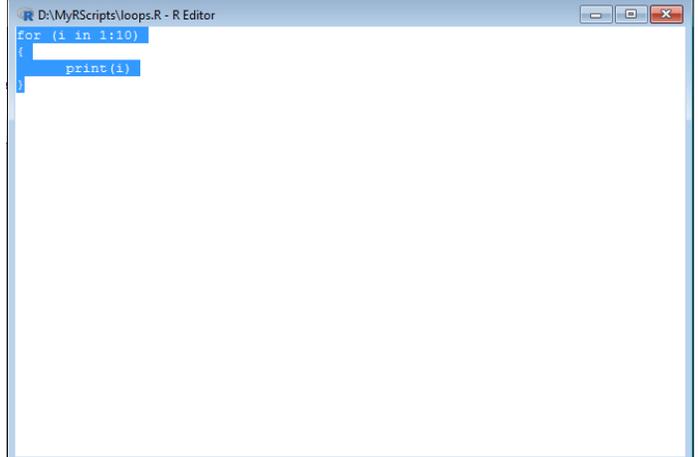


R Programming Fundamentals for Business Students— Loops

Nick V. Flor, University of New Mexico (nickflor@unm.edu)

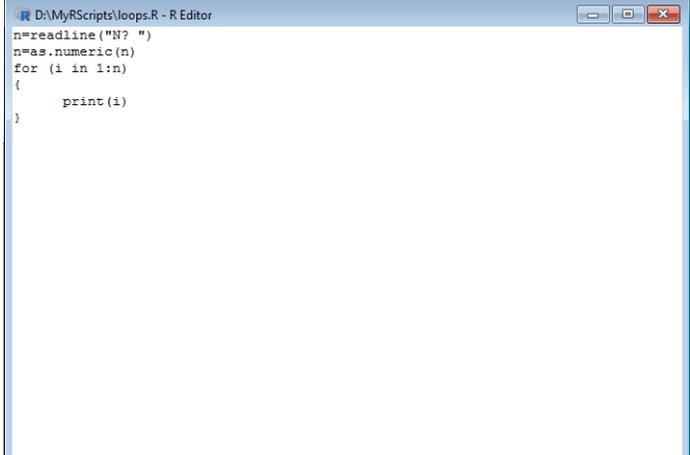
ACTION	REACTION
<ul style="list-style-type: none">○ Startup R (not shown)○ File > Change dir... to your workspace from the previous tutorial (not shown)○ File > New script (or ctrl-N) (not shown)○ File > Save as... loops.R <p>In the previous tutorial we explored conditionals in salescalc2.R. Next, we'll look at loops in R. First let's program a simple loop that counts from 1-10.</p>	 A screenshot of the R IDE. The main window shows a script editor with a blank file named 'loops.R'. A smaller window titled 'Environment & History' is open in the foreground, showing an empty environment.
FOR LOOP	
<ul style="list-style-type: none">○ Enter the following code: <pre>for (i in 1:10) { print(i) }</pre> <p>This is a basic for loop with an index variable, <i>i</i>, that counts (prints) from 1 to 10</p>	 A screenshot of the R Editor window titled 'D:\MyRScripts\loops.R - R Editor'. The code from the previous section is entered and highlighted in blue: <code>for (i in 1:10) { print(i) }</code> .
<ul style="list-style-type: none">○ File > Save (or ctrl-S)○ Enter: source("loops.R") in the R console <p>R prints out the number 1 through 10.</p> <p>We can generalize the loop by getting the ending number from the user.</p>	 A screenshot of the R Console window titled 'R Console'. It shows the command <code>> source("loops.R")</code> being executed, followed by the output of the loop: <code>[1] 1</code> , <code>[1] 2</code> , <code>[1] 3</code> , <code>[1] 4</code> , <code>[1] 5</code> , <code>[1] 6</code> , <code>[1] 7</code> , <code>[1] 8</code> , <code>[1] 9</code> , <code>[1] 10</code> , and the prompt <code>> </code> .

ASIDE: Getting the Ending Number From the User

- **Modify** the code as follows:

```
n=readline("N? ")
n=as.numeric(n)
for (i in 1:n)
{
  print(i)
}
```

Instead of hard-coding the 10, we now get a number n from the user and the loop goes from 1 to n ($1:n$)

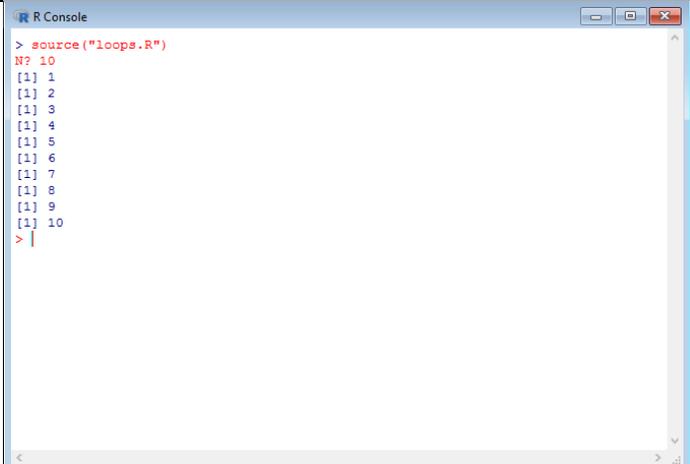


```
D:\MyRScripts\loops.R - R Editor
n=readline("N? ")
n=as.numeric(n)
for (i in 1:n)
{
  print(i)
}
```

- File > Save
- Enter: source("loops.R") in the R Console
- Enter 10 at the prompt

The code counts from 1 to whatever you enter as N .

Another common kind of loop is the while loop.



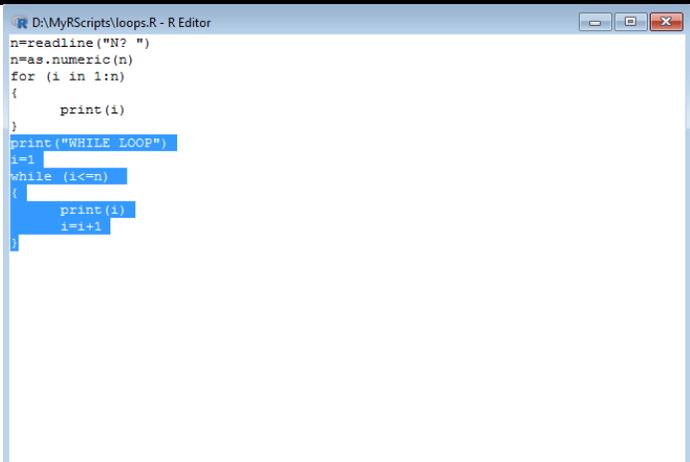
```
R Console
> source("loops.R")
N? 10
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
> |
```

WHILE LOOP

- **Add** the following code to the end of the loops.R:

```
...
print("WHILE LOOP")
i=1
while (i<=n)
{
  print(i)
  i=i+1
}
```

Note with a while loop, you have to initialize and maintain the index (i & $i=i+1$) and check the ending condition ($i<=n$)



```
D:\MyRScripts\loops.R - R Editor
n=readline("N? ")
n=as.numeric(n)
for (i in 1:n)
{
  print(i)
}
print("WHILE LOOP")
i=1
while (i<=n)
{
  print(i)
  i=i+1
}
```

- File > Save
- Enter: source("loops.R") in the R Console
- Enter 5 at the prompt (or some value)

The code counts up from 1 to 5, twice—one for the for-loop and another for the while loop.

The final loop is the repeat loop

```

R Console
> source("loops.R")
N? 5
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] "WHILE LOOP"
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> |

```

REPEAT LOOP

- Add the following code to the end of the loops.R:

```

...
print("REPEAT LOOP")
i=1
repeat {
  print(i);
  i=i+1;
  if (i>n) break;
}

```

Note with a repeat loop, you have to initialize and maintain the index (i & i=i+1) and check the ending condition (i>n), which is opposite from a while loop. You also have to explicitly break out of the loop

```

D:\MyRScripts\loops.R - R Editor
n= readline("N? ")
n= as.numeric(n)
for (i in 1:n)
{
  print(i)
}
print("WHILE LOOP")
i=1
while (i<=n)
{
  print(i)
  i=i+1
}
print("REPEAT LOOP")
i=1
repeat {
  print(i);
  i=i+1;
  if (i>n) break
}

```

- File > Save
- Enter: source("loops.R") in the R Console
- Enter 3 at the prompt (or some value)

The code counts up from 1 to 3, thrice—one for the for-loop, another for the while loop, and the last time for the repeat loop.

```

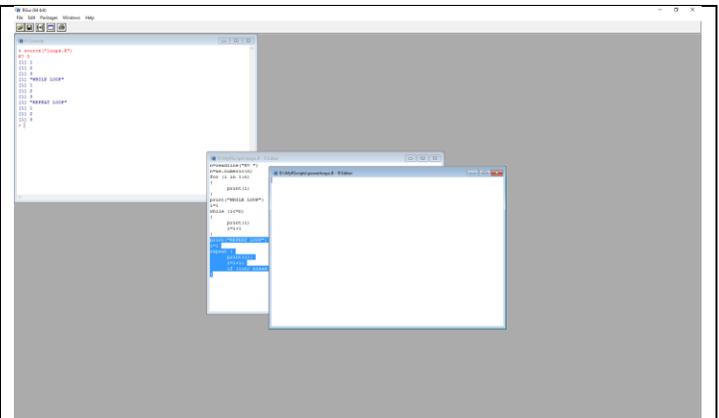
R Console
> source("loops.R")
N? 3
[1] 1
[1] 2
[1] 3
[1] "WHILE LOOP"
[1] 1
[1] 2
[1] 3
[1] "REPEAT LOOP"
[1] 1
[1] 2
[1] 3
> |

```

EXAMPLE: POWER LOOP

- File > New Script
- File > Save as..., then enter *powerloops.R* as the filename, in Save script as dialog

If you want, you can close the window for loops.R

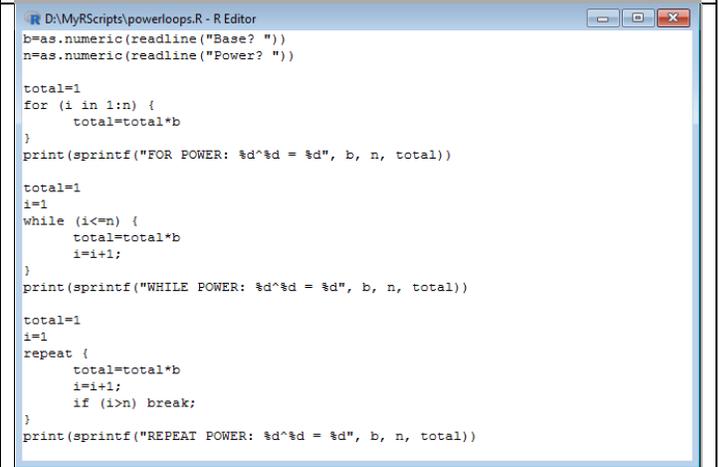


```

○ Enter the following code:
b=as.numeric(readline("Base? "))
n=as.numeric(readline("Power? "))
total=1
for (i in 1:n) {
  total=total*b
}
print(sprintf("FOR POWER: %d^%d = %d", b, n, total))

total=1
i=1
while (i<=n) {
  total=total*b
  i=i+1;
}
print(sprintf("WHILE POWER: %d^%d = %d", b, n, total))

total=1
i=1
repeat {
  total=total*b
  i=i+1;
  if (i>n) break;
}
print(sprintf("REPEAT POWER: %d^%d = %d", b, n, total))
  
```



- File > Save
- Enter: source("powerloops.R") in the R Console
- Enter 2 for Base (or some other integer)
- Enter 3 for Power (or some other integer)

The code correctly calculates base^{power} using all three different loops.

I hope that gives you a good sense of loops in R.

