Introduction to Social Data Analytics
Class 19

Arushi Kaushik

arkaushi@ucsd.edu

May 14, 2019

Today: loops

By the end of today's lecture, you should be able to:

- Construct conditional statements and for loops in `R`
- Describe how loops can reduce coding necessary to accomplish data analysis
- Define "iteration" and give examples of how the "counter" can be used within a for loop
- Recall from the Excel lectures how to use the `if` operator and describe the syntax in `R`

Open class19.R if you haven't already.

if statements in R

- The general form is:

```
if(logical test) {
  do some stuff when logical test is TRUE
}
```

- For example:

```
if(locked(door) == 1) {
  unlock(door)
}
```

- As in Stata, the action only takes place when the if statement is true.

Example in R

```
x <- sample(-1:1, 1) # random draw from {-1, 0, 1}
x
```

```
## [1] 1
```

```
if (x > 0) {
  print("x is positive")
}
```

```
## [1] "x is positive"
```

Learn this. . . or *else*!

- Use `else` to add conditions to an `if` statement:

```
if(logical test) {
  do some stuff when logical test is TRUE
} else {
  do some stuff when logical test is FALSE
}
```

- For example:

```
if(locked(door) == 1) {
  unlock(door)
  open(door)
} else {
  open(door)
}
```

Example in R

```
x <- sample(-1:1, 1)
x
```

```
## [1] -1
```

```
if (x > 0) {
  print("x is positive")
} else {
  print("x is negative")
}
```

```
## [1] "x is negative"
```

What about zero?

```
x <- sample(-1:1, 1)
x
```

```
## [1] 0
```

```
if (x > 0) {
  print("x is positive")
} else if (x < 0) {
  print("x is negative")
} else {
  print("x is zero")
}
```

```
## [1] "x is zero"
```

Takeaways from combining `if` and `else`

- You do not specify the criterea for the `else` part
- These statements are *exhaustive*: they check all possible cases
- Make sure the `else` shows up on the same line as the prior closed bracket

What is a loop?

- A sequence of commands to be repeated
- You do various sorts of loops in your daily life

```
for(i in 1:length(homework)){
  do(homework[i])
  homework[i] <- 0
}
if(sum(homework)) == 0 {
  watch(Netflix)
}
```

- In R, they are helpful for tasks you would like repeated, where typing repeatedly would be cumbersome.

for Loops in R

- The general form is:

```
for(some set of things){
  do some stuff
}
```

- For example:

```
for(i in c(keys, phone, wallet)){
  check.on.person(i)
}
```

- The loop iterates through multiple rounds.
- How many iterations in the example?

Example in R

```
x <- 0
for(i in 1:5){
  x <- x + 2*i
  print(x)
}
```

```
## [1] 2
## [1] 6
## [1] 12
## [1] 20
## [1] 30
```

What is the value of x after the fifth iteration of the loop?

Your turn: practice conditional statements and loops

Words of wisdom:

- Loops are in general slow to compute
- Faster to use vector/matrix operations
- Try each problem in class19.R with loop solutions to start

Here are the commands/operators we covered today:

- `for, if, else`
- `in, print, sample`