

Introduction to Social Data Analytics

Class 20

Arushi Kaushik

arkaushi@ucsd.edu

May 16, 2019

Today: more practice with loops

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

- ▶ Build loops that utilize the 'counter' (e.g. `i`) in three ways:
 - ▶ As a number for calculations
 - ▶ As a subset index
 - ▶ As an element of a vector (of strings or numbers)
- ▶ Construct `while` loops that may include conditional statements

Open `class20.R` if you haven't already.

Tools in your loop toolbox

- ▶ `for`, `if`, `else`, and (soon) `while`
- ▶ Using `i` as a number itself
- ▶ Using `i` to subset data
- ▶ Using `i` as an element of a vector

The problem: no year vector

From pre class exercise:

- ▶ Suppose I have data on the US population from 1790 - 1970
- ▶ But I don't have a year vector

```
data(uspop)
summary(uspop)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      3.93   15.00   50.20   69.77  114.25  203.20
```

```
n <- length(uspop)
```

- ▶ How could I generate a year vector using loops?

Option 1: use i as a number itself

```
for(i in 1:n){  
  print(1780 + 10*i)  
}
```

Option 1: use i as a number itself

```
for(i in 1:n){  
  print(1780 + 10*i)  
}
```

```
## [1] 1790  
## [1] 1800  
## [1] 1810  
## [1] 1820  
## [1] 1830  
## [1] 1840  
## [1] 1850  
## [1] 1860  
## [1] 1870  
## [1] 1880  
## [1] 1890  
## [1] 1900
```

Option 2: use i to subset data

```
year <- rep(NA, n)
for(i in 1:n){
  year[i] <- 1780 + 10*i
  print(year[i])
}
```

Option 2: use i to subset data

```
year <- rep(NA, n)
for(i in 1:n){
  year[i] <- 1780 + 10*i
  print(year[i])
}
```

```
## [1] 1790
## [1] 1800
## [1] 1810
## [1] 1820
## [1] 1830
## [1] 1840
## [1] 1850
## [1] 1860
## [1] 1870
## [1] 1880
```


Option 3: `i` as an element of a vector

```
year <- seq(1790, 1970, 10)
for(i in year){
  print(i)
}
```

- This use is useful when `i` refers to an element of a vector of text strings

Option 3: i as an element of a vector

```
year <- seq(1790, 1970, 10)
for(i in year){
  print(i)
}
```

```
## [1] 1790
## [1] 1800
## [1] 1810
## [1] 1820
## [1] 1830
## [1] 1840
## [1] 1850
## [1] 1860
## [1] 1870
## [1] 1880
## [1] 1890
```

Third use with a vector of text strings

```
instructors <- c("Arushi", "Cameron", "Duy",  
                "Mitch", "Zack")  
for(i in instructors){  
  print(paste(i,"is an instructor."))  
}
```

```
## [1] "Arushi is an instructor."  
## [1] "Cameron is an instructor."  
## [1] "Duy is an instructor."  
## [1] "Mitch is an instructor."  
## [1] "Zack is an instructor."
```

An example with for, if, and else

- ▶ For each number i in 1:10, output whether i is even.
- ▶ To do this, we will use if statements within a for loop.
- ▶ First, let's introduce `%%` and `paste()`:

```
7 %% 5
```

```
## [1] 2
```

```
remainder <- 7 %% 5  
paste(remainder, "is the remainder")
```

```
## [1] "2 is the remainder"
```

An example with `for`, `if`, and `else`

Step 1:

- ▶ Write the `if` statements needed to test whether a number `i` is even
- ▶ If it is even, output the number and “is even.”
- ▶ If it is odd, output the number and “is odd”

An example with for, if, and else

```
if(i %% 2 == 0){  
  print(paste(i,"is even."))  
} else {  
  print(paste(i,"is odd."))  
}
```

- Now do Step 2: add the for loop so that the commands above are performed over all values in 1:10

An example with for, if, and else

```
for(i in 1:10){  
  if(i %% 2 == 0){  
    print(paste(i,"is even."))  
  } else {  
    print(paste(i,"is odd."))  
  }  
}
```

An example with for, if, and else

```
for(i in 1:10){  
  if(i %% 2 == 0){  
    print(paste(i,"is even."))  
  } else {  
    print(paste(i,"is odd."))  
  }  
}
```

```
## [1] "1 is odd."  
## [1] "2 is even."  
## [1] "3 is odd."  
## [1] "4 is even."  
## [1] "5 is odd."  
## [1] "6 is even."  
## [1] "7 is odd."  
## [1] "8 is even."
```


while Loops in R

- ▶ The general form is:

```
while(logical test){  
  do some stuff until logical test becomes false  
}
```

- ▶ For example:

```
while(money > $0){  
  eat(food)  
  money <- money - cost(food)  
}
```

- ▶ When does the loop end? What happens if food is free?

Example in R

```
x <- 0
limit <- 10
while(x <= limit){
  x <- x + 2
  print(x)
}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
```

How many iterations does this loop go through?

An example with while, if, and else

```
i <- 1
while(i <= 10){
  if(i %% 2 == 0){
    print(paste(i,"is even."))
  } else {
    print(paste(i,"is odd."))
  }
  i <- i + 1
}
```

```
## [1] "1 is odd."
## [1] "2 is even."
## [1] "3 is odd."
## [1] "4 is even."
## [1] "5 is odd."
## [1] "6 is even."
```

Your turn!

- ▶ Work on class20.R and/or class19.R.

Here are the commands/operators we covered today:

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