Bootcamp_Exercise1

Regina Zweng

9/22/2017

Exercise 1) Write a for loop statements so that it runs from 1:9 and prints the following output to your screen:

-

```
for(ii in 1:9){
  if (ii < 9) {
    cat("\n")
    }
    else{
      cat("*")
      }
    }
```

```
##
##
##
##
##
##
##
##
## *
```

Exercise 2) Modify your for loop so that it prints 10 asterisks, with each asterisk separated by exactly one ampersand sign (&), with no spaces or new line characters.

```
for(ii in 1:10){
  if (ii < 10) {
    cat("*&")
    }
    else{
      cat("*")
      }
    }
```

```
## *&*&*&*&*&*&*&*&*&*
```

Exercise 3) by hand, figure out the initial values of these variables and values at the the start and end of each iteration of the loop

```
dogs <- 10;
for (i in 1:5){
    dogs <- dogs + 1;
}
#Initial value is 10, final value is 15
```

```
###
meatloaf <- 0;
for (i in 5:9){
    meatloaf <- meatloaf - i + 1;
    cat(meatloaf)
}
```

```
## -4-9-15-22-30
```

```
#Initial value is 0. Final value is -30
###
bubbles <- 12;
for (i in -1:-4){
    bubbles <- i;
}
#Initial value is 12. Final value is -4
```

Exercise 4) modify this code so that it will print out a message during presidential as well as congressional election years

```
###you can use the if statement with the modulus operator to conditionally perform operations
years <- c( 2015, 2016, 2018, 2020, 2021)
for(ii in 1:length(years)){
    if(years[ii] %% 4 == 0){
        cat(years[ii], 'Hooray, presidential and congressional elections!', sep = '\t', fill = T)
    }

    if(years[ii] %% 2018 == 0){
        cat(years[ii], 'Hooray, congressional elections!', sep = '\t', fill = T)
    }

}
```

```
## 2016
## Hooray, presidential and congressional elections!
## 2018
## Hooray, congressional elections!
## 2020
## Hooray, presidential and congressional elections!
```

Exercise 5) More fun with loops. Here are the bank accounts from seven randomly selected UCLA grad students

```
bankAccounts <- c(10, 9.2, 5.6, 3.7, 8.8, 0.5);

#Now look at the error message the following lines of code produce. Can you think of a way to modify th
compounded<-rep(bankAccounts)
interestRate <- 0.0125;
for (i in 1:length(bankAccounts)) {
    compounded[i] <- interestRate*bankAccounts[i] + bankAccounts[i]; }

#HINT: variables must be initialized before you can perform operations on them
```

```
#HINT 2: look at the rep() function and see if you can use that to initialize a variable that will help

compounded<-rep(bankAccounts)
compounded2<-rep(compounded)

for (i in 1:length(bankAccounts)) {
    compounded[i] <- interestRate*bankAccounts[i] + bankAccounts[i]; }
```

Exercise 6) Go back to the compounded interest example. Suppose we now want to compound the interest annually, but across a period of 5 years. The for loop we discussed earlier only compounds for a single year. Try this:

```
bankAccounts <- c(10, 9.2, 5.6, 3.7, 8.8, 0.5);

#Now look at the error message the following lines of code produce. Can you think of a way to modify th
bankAccounts <- c(10, 9.2, 5.6); #define bank accounts here
interestRate <- 0.0525;
house <- c(4.8, 3.8, 5.7); #deduct
food<- c(3.5, 4.3, 5.0);    #deduct
fun <- c(7.8, 2.1, 10.5);  #deduct
#and incomes (through TAships) of
income <- c(21, 21, 21); #add this
```