Bootcamp_Exercise1

Regina Zweng

9/22/2017

setwd("/Users/reginazweng/Documents/GitHub/Rbootcamp2") 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, after first loop it is 11, 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. After first loop it is -4. Final value is -30
###
bubbles <- 12;
for (i in -1:-4){
    bubbles <- i;
}
#Initial value is 12. After first loop it is -1. 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

year<-5
compounded<-matrix(NA,nrow=year,ncol=length(bankAccounts))
compounded[1,]<-bankAccounts


for (j in 2:year){
  for (i in 1:length(bankAccounts)) {
    compounded[j,i] <- interestRate*compounded[j-1,i] + compounded[j-1,i]; }
  }
compounded
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]      [,6]
## [1,] 10.00000 9.200000 5.600000 3.700000 8.800000 0.5000000
## [2,] 10.12500 9.315000 5.670000 3.746250 8.910000 0.5062500
## [3,] 10.25156 9.431437 5.740875 3.793078 9.021375 0.5125781
## [4,] 10.37971 9.549330 5.812636 3.840492 9.134142 0.5189854
## [5,] 10.50945 9.668697 5.885294 3.888498 9.248319 0.5254727
```

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); #define bank accounts here
interestRate <- 0.0525;
house <- c(4.8, 3.8, 5.7); #deduct
housematrix <-matrix(house,nrow=5,ncol=length(house),byrow=TRUE)
food<- c(3.5, 4.3, 5.0);    #deduct
foodmatrix <-matrix(food,nrow=5,ncol=length(food),byrow=TRUE)
fun <- c(7.8, 2.1, 10.5);  #deduct
funmatrix <-matrix(fun,nrow=5,ncol=length(fun),byrow=TRUE)
#and incomes (through TAships) of
income <- c(21, 21, 21); #add this
incomematrix <-matrix(income,nrow=5,ncol=length(income),byrow=TRUE)

bankAccountsMod<-bankAccounts-house-food-fun+income
year<-5
compounded<-matrix(NA,nrow=year,ncol=length(bankAccountsMod))
compounded[1,]<-bankAccountsMod
compounded
```

```
##      [,1] [,2] [,3]
## [1,] 14.9   20  5.4
## [2,]   NA   NA   NA
## [3,]   NA   NA   NA
## [4,]   NA   NA   NA
## [5,]   NA   NA   NA
```

```
for (j in 2:year) {
      for (i in 1:length(bankAccounts)) {
        compounded[j,i]<- compounded[j-1,i]-housematrix[j-1,i]-foodmatrix[j-1,i]-funmatrix[j-1,i]+incom
      }
        {
    compounded[j,i] <- interestRate*compounded[j-1,i] + compounded[j-1,i]; }
  }
compounded
```

```
##      [,1] [,2]     [,3]
## [1,] 14.9 20.0 5.400000
## [2,] 19.8 30.8 5.683500
## [3,] 24.7 41.6 5.981884
## [4,] 29.6 52.4 6.295933
## [5,] 34.5 63.2 6.626469
```

Exercise 7) Three students have estimated annual expenditures for food, housing, and fun of: (in thousands of dollars) Modify the 5-year interest-compounding code from #5 and #6 so that it runs from 2015-2020 and so that in odd numbered years students 1 and 3 get trust fund disbursements of $5000. (hint the modulus function %% will be helpful)

```
bankAccounts <- c(10, 9.2, 5.6); #define bank accounts here
interestRate <- 0.0525;
house <- c(4.8, 3.8, 5.7); #deduct
housematrix <-matrix(house,nrow=5,ncol=length(house),byrow=TRUE)
foodmod<- c(3.5, 4.3, 5.0);    #deduct
foodmatrix <-matrix(food,nrow=5,ncol=length(food),byrow=TRUE)
fun <- c(7.8, 2.1, 10.5);  #deduct
funmatrix <-matrix(fun,nrow=5,ncol=length(fun),byrow=TRUE)
#and incomes (through TAships) of
income <- c(21, 21, 21); #add this
incomematrix <-matrix(income,nrow=5,ncol=length(income),byrow=TRUE)
trustfund<-c(5,0,5)
trustfundmatrix<-matrix(trustfund,nrow=5,ncol=length(trustfund),byrow=TRUE)
trustfundmatrix[2,]<-c(0,0,0)
trustfundmatrix[4,]<-c(0,0,0)
trustfundmatrix
```

```
##      [,1] [,2] [,3]
## [1,]    5    0    5
## [2,]    0    0    0
## [3,]    5    0    5
## [4,]    0    0    0
## [5,]    5    0    5
```

```
bankAccountsMod<-bankAccounts-house-food-fun+income+trustfund
year<-5
compounded<-matrix(NA,nrow=year,ncol=length(bankAccountsMod))
compounded[1,]<-bankAccountsMod
compounded
```

```
##      [,1] [,2] [,3]
```

```
## [1,] 19.9   20 10.4
## [2,]   NA   NA   NA
## [3,]   NA   NA   NA
## [4,]   NA   NA   NA
## [5,]   NA   NA   NA
```

```
for (j in 2:year) {
      for (i in 1:length(bankAccounts)) {
        compounded[j,i]<- compounded[j-1,i]-housematrix[j-1,i]-foodmatrix[j-1,i]-funmatrix[j-1,i]+incom
      }
        {
    compounded[j,i] <- interestRate*compounded[j-1,i] + compounded[j-1,i]; }
  }
compounded
```

```
##      [,1] [,2]     [,3]
## [1,] 19.9 20.0 10.40000
## [2,] 24.8 30.8 10.94600
## [3,] 34.7 41.6 11.52066
## [4,] 39.6 52.4 12.12550
## [5,] 49.5 63.2 12.76209
```

Exercise 8) use a while loop to sum all numbers from 1:17. You will need to use a counter variable (like index seen in class).

```
 j <- 0
 i <- 1
while(i<=17){
   j <- j+i
   i <- i+1
}
j
```

```
## [1] 153
```

Exercise 9) write a function takes a number, and prints 'small' if number less than or equal to -1; 'medium' if between -1 and + 1'big' if greater than or equal to + 1

```
size<-function(num){
  if (num <= -1) cat('small');
  if (-1< num & num <1) cat('medium');
  if (num >= 1) cat('big');}
#examples
size(-6)
```

```
## small
```

```
size(-1)
```

```
## small
```

```r
size(0)
```

```
## medium
```

```r
size(1)
```

```
## big
```

```r
size(2)
```

```
## big
```