Bootcamp Exercise 1

Madeline Cowen

September 14, 2016

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

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

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

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

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

3. dogs(i=1) = 11
   dogs(i=5) = 15
   ###
   i=5
   meatloaf = -4
   output = -4
   i=9
   meatloaf = -30
   output = -4-9-15-22-30
   ###
   i=-1

bubbles = -1
i=-4
bubbles = -4

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] %% 2 == 0){
        cat(years[ii], 'Hooray, congressional elections!', sep = '\t', fill = T)
    }


}
```

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

5. Fix the code to remove the error:

```
bankAccounts <- c(10, 9.2, 5.6, 3.7, 8.8, 0.5);
compounded <- rep(NA,length(bankAccounts))
interestRate <- 0.0125;
for (i in 1:length(bankAccounts)) {
    compounded[i] <- interestRate*bankAccounts[i] + bankAccounts[i]; }
compounded
```

```
## [1] 10.12500  9.31500  5.67000  3.74625  8.91000  0.50625
```

6.

```
bankAccounts <- c(10, 9.2, 5.6); #define bank accounts here
compounded <- rep(NA,length(bankAccounts))
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

for (j in 1:5) {
    for (i in 1:length(bankAccounts)) {
      bankAccounts[i] <- bankAccounts[i] - house[i] - food[i] - fun[i] + income[i]
        #step 1 modify bankAccounts so that amounts reflect income and expenses
```

```
        #step 2 get calculate interest and add to accounts from step 1
        #you can actually use the line you have already written if you
        #modify amounts in bankAccounts directly in step 1
      compounded[i] <- interestRate*bankAccounts[i] + bankAccounts[i]
    }
}
compounded
```

```
## [1] 36.31125 66.51800  4.84150
```

7. 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.

```
bankAccounts <- c(10, 9.2, 5.6); #define bank accounts here
compounded <- rep(NA,length(bankAccounts))
interestRate <- 0.0525;
house <- c(4.8, 3.8, 5.7);
food<- c(3.5, 4.3, 5.0);
fun <- c(7.8, 2.1, 10.5);
income <- c(21, 21, 21);

years <- seq(2015,2020,by=1)
for (j in years) {
    for (i in 1:length(bankAccounts)) {
      if (j %% 2 == 1 && i %% 2 == 1) {
        bankAccounts[i] <- bankAccounts[i] - house[i] - food[i] - fun[i] + income[i] + 5000
      }
      else {
        bankAccounts[i] <- bankAccounts[i] - house[i] - food[i] - fun[i] + income[i]
      }
        compounded[i] <- interestRate*bankAccounts[i] + bankAccounts[i]
    }
}
compounded
```

```
## [1] 15828.969    77.885 15792.131
```

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).

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

```
## [1] 133
```

9. Write a function that 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

```r
categorizer <- function(number) {
  if (number <= -1) {
    cat("small")
  }
  if (number >= 1) {
    cat("big")
  }
  if (number <1 && number > -1) {
    cat("medium")
  }
}
```