HW9b

Autumn Li

12/4/2016

1. Write a function adopters which takes two arguments, month, with no default value and not.yet defaulting to FALSE.

```
nodes = read.csv("~/Desktop/ckm_nodes-1.csv")
date = nodes$adoption_date
not.na = na.omit(date)
ind_na = which(is.element(date=="NA",date))
vect = 1:246
dr_ind = vect[-ind_na]
dim(nodes)
```

```
## [1] 246  13
```

```
nodes = nodes[dr_ind,]
date1 = nodes$adoption_date
dim(nodes)
```

```
## [1] 125  13
```

```
adopters = function(month, not.yet = FALSE){
  count = NULL
  if(not.yet == FALSE){
    for(i in 1: length(date1)){
      if(date1[i]==month){
        count[i] = 0
        }
      }
    }
  if(not.yet == TRUE){
    for(i in 1:length(date1)){
      if(date1[i]> month){
        count[i] = 0
      }
      }
    }
  result = length(na.omit(count))
  return(result)
  }
adopters(2)
```

```
## [1] 9
```

```
adopters(month = 14, not.yet = TRUE)
```

```
## [1] 23
```

2. Create a vector which stores the number of contacts each doctor has. Do not use a loop. Check that doctor number 41 has 3 contacts.

```
ckm_network.1 = read.table("~/Desktop/ckm_network-1.txt", quote="\"")
ckm_txt = ckm_network.1
dim(ckm_txt)
```

```
## [1] 246 246
```

```
network = ckm_txt[-ind_na,-ind_na]
num_contact = apply( network , 1 , sum)
num_contact =  as.vector( num_contact )
num_contact[41]
```

```
## [1] 3
```

3.

```
nodes_work = nodes
count_peer_pressure = function( index , month ){
  nodes_work$contact = as.numeric(network[index , ] )
  nodes_work_2 = subset( nodes_work , ( nodes_work$adoption_date <= month ) & (nodes_work$contact == 1)
  return(dim(nodes_work_2)[1])
}
count_peer_pressure ( 37, 5)
```

```
## [1] 3
```

4.

```
prop_peer_pressure = function ( index , month){
  result = count_peer_pressure( index , month)/sum(network[ , index])
  return(result)
}
prop_peer_pressure( 37 , 5 )
```

```
## [1] 0.6
```

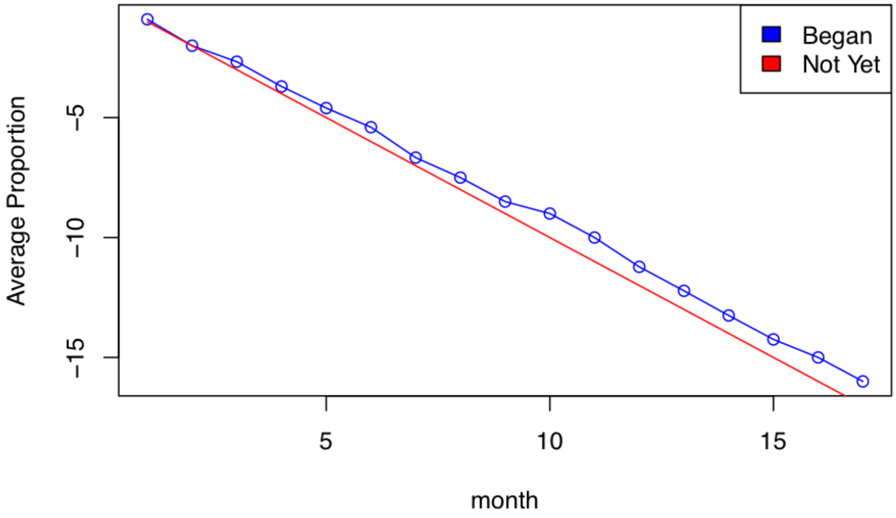
5.

```
func = function(m){
  x = adopters(m)
  y = adopters(m,T)
  a = mean(sapply(x,prop_peer_pressure,month = m),na.rm = T)
  b = mean(sapply(y,prop_peer_pressure,month = m),na.rm = T)
  return(c(a,b))
}
func(5)
```

```
## [1] 0.4 0.4
```

6.

```r
par(mfrow=c(1,1))
x = 1:17
y = sapply(x,func)
plot(y[1,]-x,type = "o",col = "blue",xlab = "month",ylab = "Average Proportion")
lines(y[2]-x,col = "red")
legend("topright",legend=c("Began","Not Yet"),fill = c("blue","red"))
```


🟦 Began
🟥 Not Yet
Average Proportion
month