Homework 9 Solutions

Cynthia Rush (cgr2130)

Dec 5, 2016

Your homework should be submitted on Canvas as an `R` Markdown file. Please submit the knitted .pdf file along with the .Rmd file. We will not accept any other formats. Please clearly label the questions in your responses and support your answers by textual explanations and the code you use to produce the result. Note that you cannot answer the questions by observing the data in the "Environment" section of `R`Studio or in Excel – you must use coded commands. We will print out your homeworks. Please do not waste paper by printing the dataset or any vector over, say, length 20.

In this homework you'll explore transforming data and get more practice with selective access and applying functions.

Ideas which we take today as common such as `viral marketing' and`early adopters' grew from sociological studies on the diffusion of information. One of the most famous such studies tracked how a then-new antibiotic, tetracycline, spread among doctors in four small cities in Illinois in the 1950s. In this lab we will study this data and study the idea that the innovation (in this case tetracycline) 'spread' from one person to the next.

Download the two data files `ckm_nodes.csv` and `ckm_network.txt` which store information on each individual doctor in the four towns and on which doctors knew each other, respectively.

Part I

1. Load the `ckm_nodes.csv` data into a data frame called `nodes`. It should have 246 rows and 13 columns. The variable `adoption_date` records the month in which the doctor began prescribing tetracycline, counting from November 1953. If the doctor did not begin prescribing it by month 17, i.e. February 1955, when the study ended, this is recorded as `Inf`. If it's not known when or if the doctor adopted tetracycline, their value is `NA`. Answer the following. (a) How many doctors *began* prescribing tetracycline in each month of the study? (b) How many never prescribed it? (c) How many are NAs?

```
nodes <- read.csv("ckm_nodes.csv", as.is = TRUE)
dim(nodes)
```

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

```
table(nodes$adoption_date)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17 Inf
##  11   9   9  11  11  11  13   7   4   1   5   3   3   4   4   2   1  16
```

```
sum(is.na(nodes$adoption_date))
```

```
## [1] 121
```

The table command counts the doctors who began to prescribe tetracycline in each month and it also tells us that 16 doctors never prescribed it during the duration of the study. There are 121 doctors for which we don't know this info.

2. Create a vector which records the *index numbers* of the doctors for whom `adoption_date` is not `NA`. Check that this vector has length 125. Reassign `nodes` so it only contains those rows. (Do not drop rows if they have a value for `adoption_date` but are `NA` in some other column.) Use this cleaned version of `nodes` for the rest of the homework.

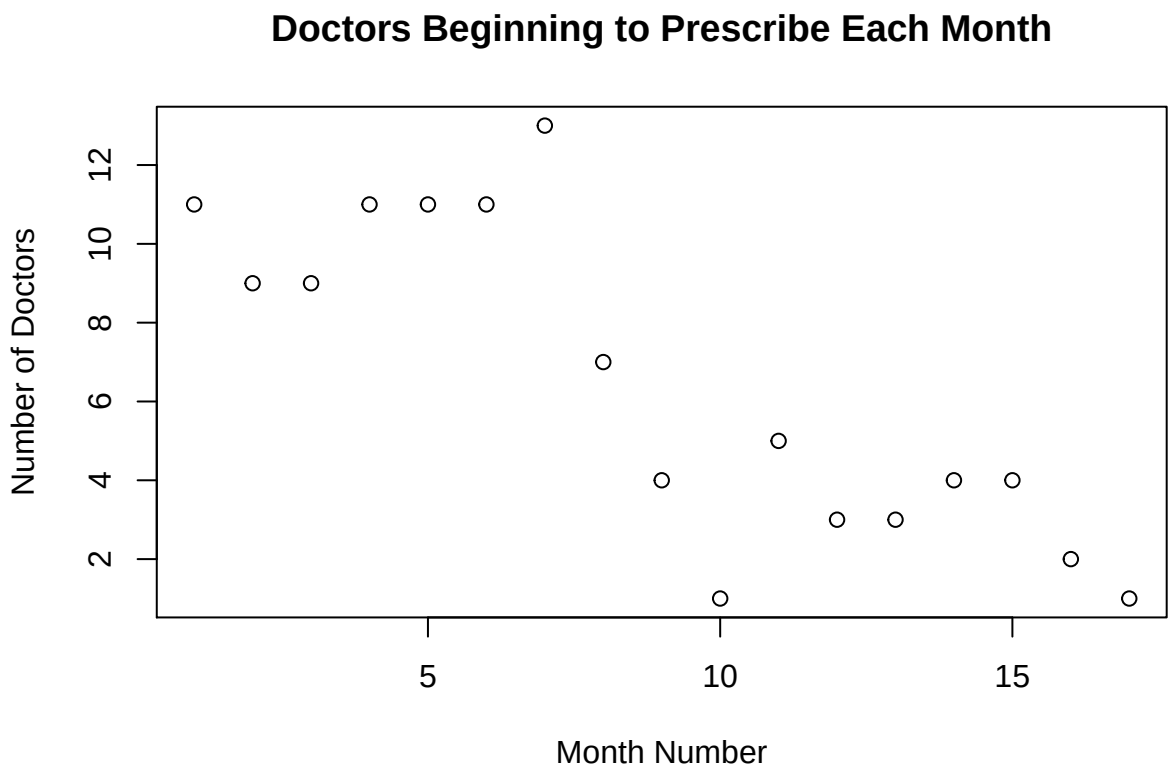
```
NAdocs <- which(is.na(nodes$adoption_date))
length(NAdocs)
```

```
## [1] 121
```

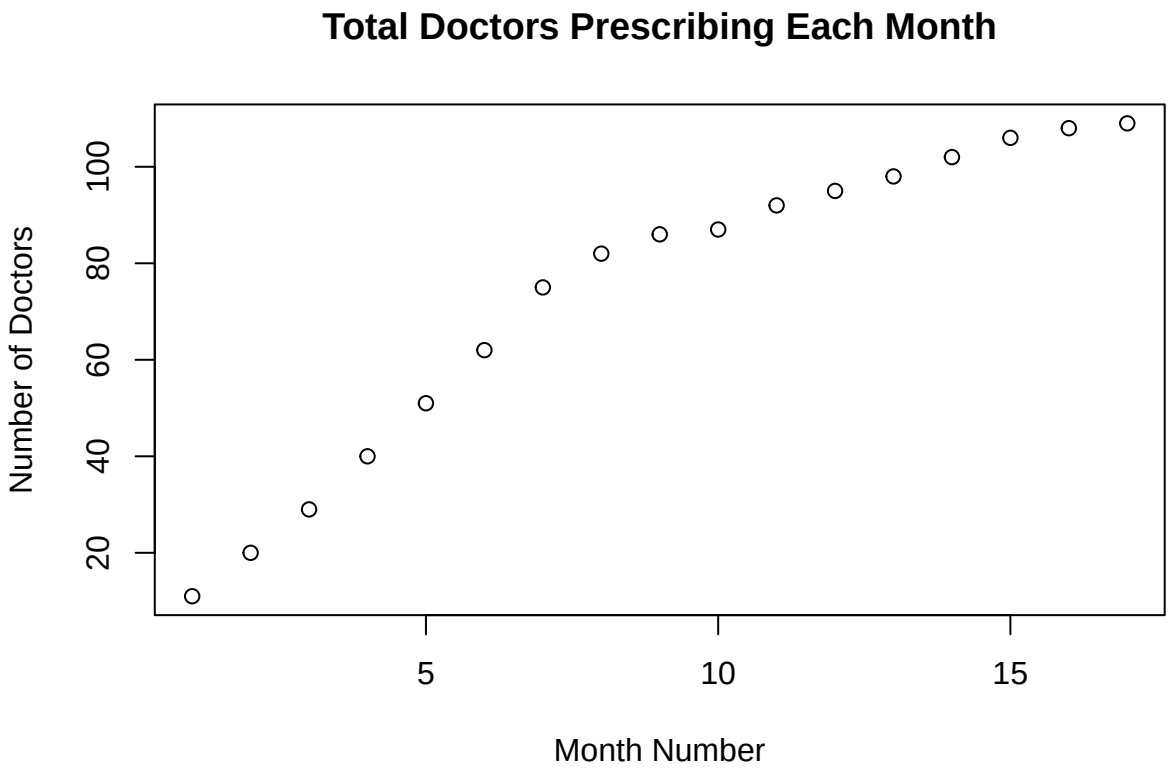
```
nodes  <- nodes[-NAdocs, ]
```

3. Create a plot of the number of doctors who began prescribing tetracycline each month versus time. (The number on the x-axis can be integers instead of formatted dates.) Produce another plot of the *total* number of doctors prescribing tetracycline in each month. The curve for total adoptions should first rise rapidly and then level out around month 6.

```
month.info <- table(nodes$adoption_date)[1:17]
plot(names(month.info), month.info, xlab = "Month Number", ylab = "Number of Doctors", main = "Doctors
```




```
plot(names(month.info), cumsum(month.info), xlab = "Month Number", ylab = "Number of Doctors", main = "
```



4. Create a logical vector which indicates for each doctor whether they had begun prescribing tetracycline by month 2. Convert it to a vector of index numbers. There should be twenty such doctors. Create a Boolean vector which indicates for each doctor whether they began prescribing tetracycline after month 14. Convert it to a vector of index numbers. There should be twenty-three such doctors.

```
month2 <- nodes$adoption_date <= 2
head(month2)
```

```
## [1]  TRUE FALSE FALSE FALSE FALSE FALSE
```

```
month2 <- which(month2)
head(month2)
```

```
## [1]  1 10 13 20 27 45
```

```
length(month2)
```

```
## [1] 20
```

```
month14 <- nodes$adoption_date > 14
head(month14)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
month14 <- which(month14)
head(month14)
```

```
## [1]  7 14 16 17 30 39
```

```
length(month14)
```

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

5. Write a function `adopters` which takes two arguments, `month`, with no default value and `not.yet` defaulting to `FALSE`. If `not.yet` is `FALSE`, `adopters` should return a logical value, indicating the doctors who began prescribing tetracycline in that month. If `not.yet` is `TRUE`, then `adopters` should return the vector indicating the doctors who began prescribing tetracycline `after` that month (or never did). Check that `adopters(2)` indicates 9 doctors began prescribing in month 2, and that `adopters(month = 14, not.yet = TRUE)` indicates that 23 doctors began prescribing after month 14, or never did.

```
adopters <- function(month, not.yet = FALSE) {
  if (not.yet) {
    return(which(nodes$adoption_date > month))
  } else {
    return(which(nodes$adoption_date == month))
  }
}
length(adopters(2))
```

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

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

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

Part II

6. The file `ckm_network.txt` contains a binary matrix; the entry in row i, column j is 1 if doctor number i said that doctor j is a friend or close professional contact, and 0 otherwise. Load the file into `R` and call it `network`. Verify that gives you a square matrix which contains only 1s and 0s with 246 rows and columns. Drop the rows and columns corresponding to doctors with missing `adoption_date` values. Check that the result has 125 rows and columns. Use this reduced matrix, and its rows and column numbers for the rest of the homework.

```
network <- read.table("ckm_network.txt", as.is = TRUE, header = FALSE)
dim(network)
```

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

```
network <- network[-NAdocs, -NAdocs]
dim(network)
```

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

7. 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.

```
doc.contacts <- rowSums(network)
doc.contacts[41]
```

```
## 70
##  3
```

8. Create a Boolean vector which indicates, for each doctor, whether they were contacts of doctor 37, *and* had begun prescribing tetracycline by month 5. Count the number of such doctors without converting the Boolean vector to a vector of indices. There should be three such doctors. What proportion of doctor 37's friends do those three doctors represent?

```
contact37 <- as.logical(network[, 37]) & nodes$adoption_date <= 5
sum(contact37)
```

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

```
sum(contact37)/doc.contacts[37]
```

```
##  37
## 0.6
```

The proportion is 0.6.

9. Write a function `count_peer_pressure` which takes in the index number of a doctor and a month and returns the number of doctors whom that doctor names as contacts, *and* had begun prescribing tetracycline by that month or earlier. If it is working properly, doctor number 37 and month 5 should return 3.

```
count_peer_pressure <- function(index, month) {
  contacts <- network[, index] & nodes$adoption_date <= month
  return(sum(contacts))
}
count_peer_pressure(37, 5)
```

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

10. Write a function `prop_peer_pressure` which takes in the index number of a doctor and a month and returns the proportion of the doctor's contacts who are already prescribing tetracycline by that month. If a doctor has no contacts, your function should return `NaN`. Check that doctor 37, month 5 returns a proportion of 0.6, but doctor 102 in month 14 returns `NaN`. Your function should call, not repeat, your function from (8) and use your vector from (6).

```
prop_peer_pressure <- function(index, month) {
  if (doc.contacts[index] == 0) {
    return(NaN)
  } else {
    return(count_peer_pressure(index, month)/doc.contacts[index])
  }
}
prop_peer_pressure(37, 5)
```

```
##  37
## 0.6
```

```
prop_peer_pressure(102, 14)
```

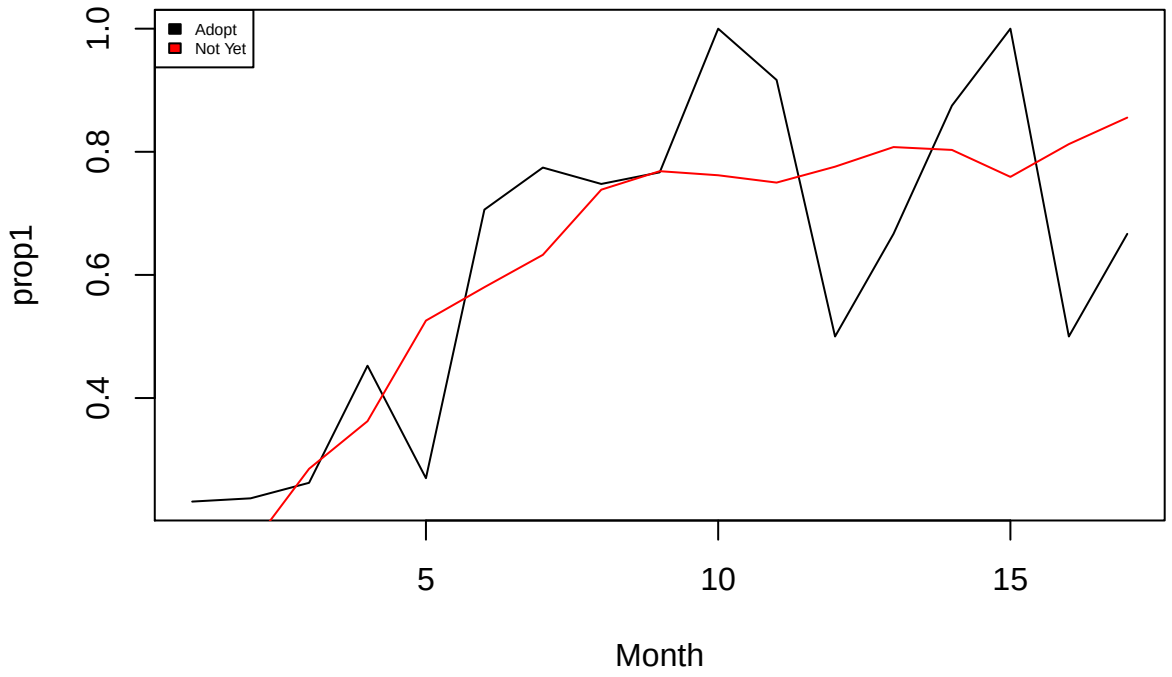
```
## [1] NaN
```

11. Write a function which takes in a month and returns a vector of length 2. The first element of the returned value should be the average proportion of prescribers among contacts of doctors who `began` prescribing in that month. The other should be the average proportion of prescribers among contacts of doctors who began prescribing *later*, or never. Call your code from (4) and (9); avoid using a loop.

```
viral <- function(month) {
  v1 <- mean(sapply(adopters(month), prop_peer_pressure, month), na.rm = TRUE)
  v2 <- mean(sapply(adopters(month, not.yet = TRUE), prop_peer_pressure, month), na.rm = TRUE)
  return(c(v1, v2))
}
```

12. Plot the average proportions from (10) over time. Do not use a loop. Do the doctors who adopt in a given month consistently have more contacts who are already prescribing than non-adopters?

```
prop1  <- sapply(1:17, viral)[1,]
prop2  <- sapply(1:17, viral)[2,]
plot(1:17, prop1, type = "l", xlab = "Month")
points(1:17, prop2, type = "l", col = "red")
legend("topleft", c("Adopt", "Not Yet"), fill = c(1,2), cex = .5)
```



It's not consistent that those beginning to prescribe have more contacts than those who aren't mainly because the proportions for those beginning to prescribe each month are highly variable.