STAT GR5206 HW5_mjs2364

Mathieu Sauterey - UNI: MJS2364

29 November 2017

Goals: Practice transforming data and 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.

Part1.

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.

i. Answer the following. (a) How many doctors began prescribing tetracycline in each month of the study?

(b) How many never prescribed it during the study? (c) How many are NAs?

```
#First we load the study data
nodes <- read.csv("ckm_nodes.csv", as.is = TRUE)

#Vector of size 17, shows how many doctors began prescribing in each of 17 months
doctors <-c()
for (i in 1:17){
  doctors[i] <- sum(nodes$adoption_date==i, na.rm=TRUE)
}

#prints the vector
doctors
```

```
##  [1] 11  9  9 11 11 11 13  7  4  1  5  3  3  4  4  2  1
```

```
#Checks how many doctors never prescribed
sum(nodes$adoption_date==Inf, na.rm=TRUE)
```

```
## [1] 16
```

```
#Checks how many adoption_date are NAs
sum(is.na(nodes$adoption_date))
```

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

Therefore, 109 doctors began prescribing tetracycline during the study. The number of doctor for each of the 17 months is [11 | 9 | 9 | 11 | 11 | 11 | 13 | 7 | 4 | 1 | 5 | 3 | 3 | 4 | 4 | 2 | 1]. 16 doctors never prescribed it. There are 121 NAs.

ii. 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.

```
# Logical vector of doctors for whom adoption_date is not NA
non_na <- !is.na(nodes$adoption_date)

# Vector of index of these doctors
non_na_ind <- which(non_na)

# Prints how many such doctors there are (should return 125)
length(non_na_ind)
```

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

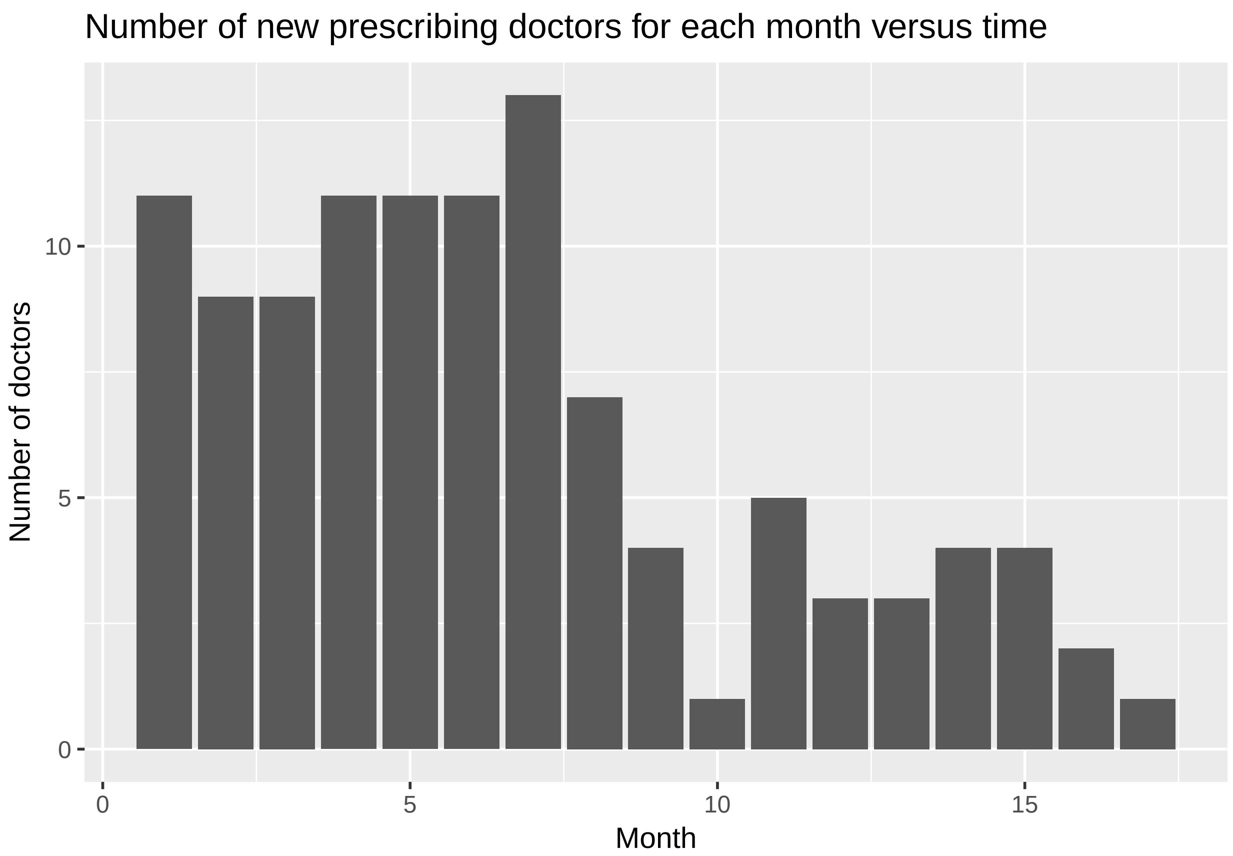
```
# Dataframe containing only these doctors without NA adoption_date
new_nodes <- subset(nodes, non_na)
```

As expected, the vector containing index numbers of non-NA adoption_date data has length 125.

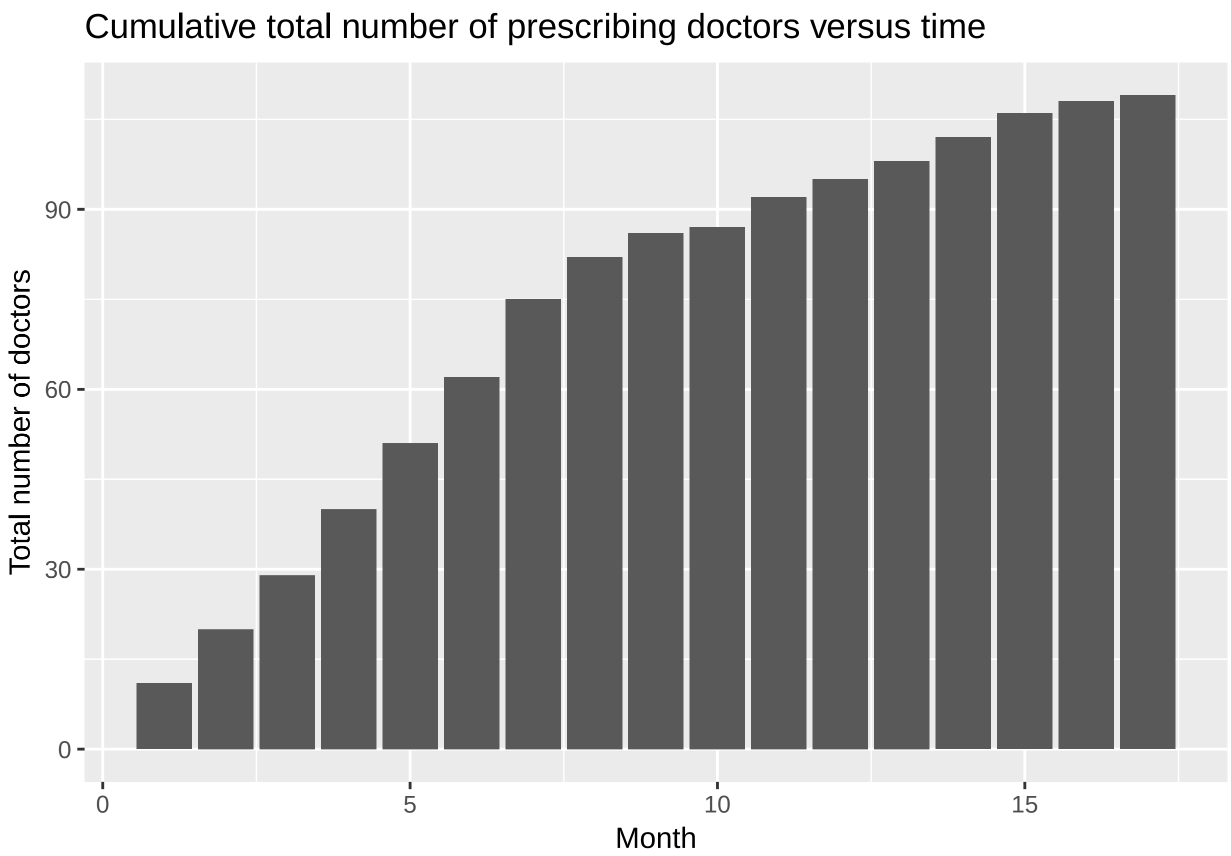
iii. Create a plot of the number of doctors who began prescribing tetracycline each month versus time. (The number on the x-axis can be integers instead of formatted dates.) Produce another plot of the total number of doctors prescribing tetracycline in each month. The curve for total adoptions should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

```
# Loads the graphic visualization package
library(ggplot2)

# Plots the number of doctor who began prescribing on each month vs time
ggplot(data.frame(nodes$adoption_date))+
  geom_bar(mapping=(aes(x=nodes$adoption_date)))+
  labs(title="Number of new prescribing doctors for each month versus time",
       x="Month", y="Number of doctors")
```

```
## Warning: Removed 137 rows containing non-finite values (stat_count).
```



```
# Facilitates the bar graph plot
total_doctors <- rep(1:17, cumsum(doctors))

# Plots the total number of doctors prescribing in each month vs time
ggplot(data.frame(total_doctors))+
  geom_bar(mapping=(aes(x=total_doctors)))+
  labs(title="Cumulative total number of prescribing doctors versus time",
       x="Month", y="Total number of doctors")
```



iv. 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 logical 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.

```
# Logical vector of docs who had begun prescribing by month 2
doc_month2 <- new_nodes$adoption_date <= 2

# Vector index of such docs
doc_month2_ind <- which(doc_month2)

# Number of docs who had begun prescribing by month 2 (should return 20)
sum(doc_month2)
```

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

```
# Logical vector of docs who began prescribing after month 14
doc_month14 <- new_nodes$adoption_date > 14

# Vector index of such docs
doc_month14_ind <- which(doc_month14)
```

```r
# Number of docs who began prescribing after month 14 (should return 23)
sum(doc_month14)
```

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

As expected, there are 20 doctors that had begun prescribing by month 2, and 23 who began prescriing after month 14.

v. 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 vector indicating the (indices of the) doctors who began prescribing tetracycline in that month. If not.yet is TRUE, then adopters should return the vector indicating the (indices of 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.

```r
# Function's description above (NEW QUESTION DESCRIPTION USING INDICES)
adopters <- function(month, not.yet=FALSE){

  if (not.yet==FALSE){
    doc_month <- which(new_nodes$adoption_date == month)
  }

  else{
    doc_month <- which(new_nodes$adoption_date > month  | new_nodes$adoption_date == Inf)
  }

  return(doc_month)
}


# Prints how many docs began prescribing in month 2 (should return 9)
length(adopters(2))
```

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

```r
# Prints how many docs began prescribing after month 14, or never (should return 23)
length(adopters(14, not.yet = TRUE))
```

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

```r
# Adopter's function returning logical (OLD QUESTION DESCRIPTION USING LOGICAL VECTOR)
adopters <- function(month, not.yet=FALSE){

  if (not.yet==FALSE){
    doc_month <- new_nodes$adoption_date == month
  }

  else{
    doc_month <- (new_nodes$adoption_date > month  | new_nodes$adoption_date == Inf)
  }
  return(doc_month)
}
```

As expected, adopters(2) correctly indicates that 9 doctors began prescribing in month 2, and adopters(month = 14, not.yet = TRUE) also correctly indicates that 23 doctors began prescribing after month 14, or never did.

Part 2

vi. 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 you get a square matrix that 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.

```
# First we load the study data
network <- matrix(scan("ckm_network.txt"),nrow=246, byrow=TRUE)

# Checks what values the files contains (should return 0 and 1)
unique(c(network))
```

```
## [1] 0 1
```

```
# Checks the dimensions of matrix network (should return 246x246)
dim(network)
```

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

```
# Drops the rows and columns with NA adoption_date
network <- network[non_na_ind,non_na_ind]

# Checks the new dimensions of matrix network (should return 125x125)
dim(network)
```

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

As expected, the dimension of 'network' is 246x246 and it is thus a squared matrix. Also, it only contains 1s and 0s. After removing from the matrix all rows and columns that have NA values for adoption_date, we correctly obtain a 125x125 matrix.

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

```
# Vector of the number of friends/contacts each doctor has
doc.contacts <- apply(network, 1, sum)

# Prints how many contacts doctor #41 has (should return 3)
doc.contacts[41]
```

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

Doctor number 41 has 3 contacts, as expected.

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

```r
#Logical vector shows if each doc is a friend of doc #37,and began prescribing <= month 5
contact_37_5 <- as.logical(network[,37]) & !adopters(month = 5, not.yet = TRUE)

# Prints how many such doctor there are
sum(contact_37_5)
```

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

```r
# Prints how many friends doc#37 has
doc.contacts[37]
```

```
## [1] 5
```

```r
# Prints the proportion of doc #37's friends who had begun prescribing <= month 5
sum(contact_37_5)/doc.contacts[37]
```

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

We correctly count 3 such doctors who were a contact of doctor #37 and had begun prescribing by month 5 or earlier. Given that doctor #37 has 5 contacts, this represents 60% of his friends.

ix. Write a function count_peer_pressure that 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.

```r
# Function's description above
count_peer_pressure <- function(index, month_input){

  mat <- (network[,index] & !adopters(month_input, not.yet = TRUE))

  if(!is.null(dim(mat))){
    return(apply(mat, 2, sum))}

  else{
    return(sum(mat))}
}

# Prints how many contacts of doc #37 had begun prescribing <= month 5 (should return 3)
count_peer_pressure(37,5)
```

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

The function works correctly because it returns 3 such doctors who were a contact of doctor #37 and had begun prescribing by month 5 or earlier.

x. Write a function prop_peer_pressure that 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 the code from, your count_peer_pressure() function and use your doc.contacts vector.

```
# Function's description above
prop_peer_pressure <- function(index, month_input){

  num_contacts <- doc.contacts[index]
  presc_contacts <- count_peer_pressure(index, month_input)

  if (sum(num_contacts)==0){
    return(NaN)}
  else{
    return(presc_contacts/num_contacts)}
}

# Prints what proportion of doc #37's contacts had begun prescribing <= month 5 (=0.6)
prop_peer_pressure(37,5)
```

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

```
# Prints what proportion of doc #102's contacts had begun prescribing <= month 14 (=NaN)
prop_peer_pressure(102,14)
```

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

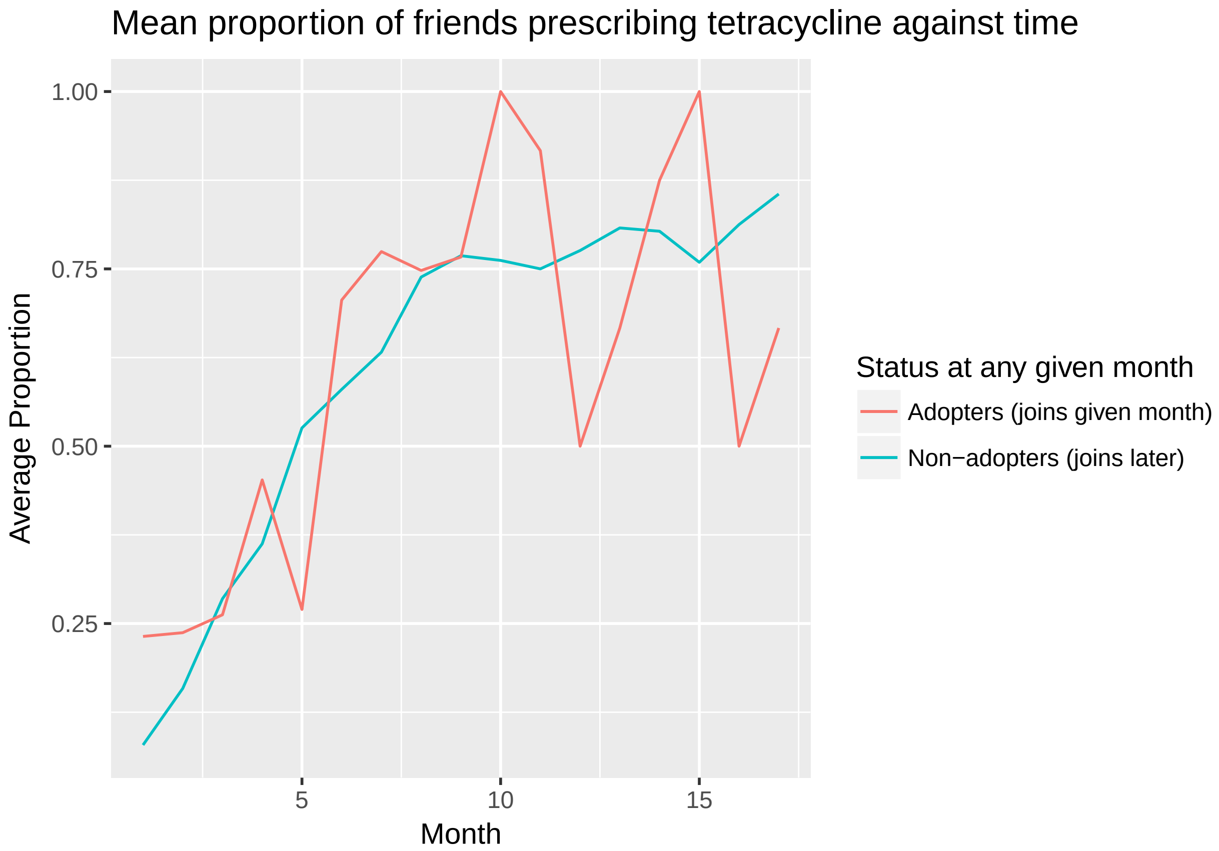
The function works correctly because it returns 3 such doctors who were a contact of doctor #37 and are already prescribing by month 5, and it returns NaN for doctor #102 in month 14.

xi. Write a function that takes in a month and returns a vector of length 2. The first element of the returned vector 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 adopters() and prop_peer_pressure() functions; avoid using a loop by using an appropriate function from the apply family.

```
# Function's description above
proportion <- function(month_input){

  ind_doc        <- which(adopters(month_input)==TRUE)
  ind_doc_later  <- which(adopters(month_input, not.yet = TRUE))
  prop_doc       <- prop_peer_pressure(ind_doc,month_input)
  prop_doc_later <- prop_peer_pressure(ind_doc_later,month_input)

  vector <- c(NA,NA)
  vector[1] <- mean(prop_doc, na.rm = TRUE)
  vector[2] <- mean(prop_doc_later, na.rm = TRUE)
  return(vector)
}
```

xii. Plot the average proportions from (11.) over time on the same graph. Do not use a loop and add an appropriate legend. The point of the plot is to visualize whether the doctors who adopt in a given month consistently have more contacts who are already prescribing than non-adopters.

```
# Creates dataframe of proportions (one column for each two types, and 17 rows of months)
ave_prop <- t(sapply(1:17, proportion))

#Renames each column to indicate if the mean proportions are for the given month or later
colnames(ave_prop) <- c("Month","Later_Months")

# Plots mean proportions for each type, against each of the 17 months
ggplot(data.frame(ave_prop), colour=Type)+
  geom_line(mapping = aes(x = 1:17, y = Later_Months, col="Non-adopters (joins later)"))+
  geom_line(mapping = aes(x = 1:17, y = Month, col="Adopters (joins given month)" ))+
  scale_color_discrete(name = "Status at any given month")+
  labs(title="Mean proportion of friends prescribing tetracycline against time",
      x="Month", y="Average Proportion")
```



Based on the plot, we do observe that the doctors who start prescribing (adopters) in a given month do not consistently have more contacts who are already prescribing than non-adopters.