Problem Set 3

Martin Aragoneses

July 23, 2015

Data

The data for this assignment come from the Hospital Compare web site (http://hospitalcompare.hhs.gov) run by the U.S. Department of Health and Human Services, providing information about the quality of care at over 4,000 Medicare-certified hospitals in the U.S. This dataset essentially covers all major U.S. hospitals. There are three files:

- `outcome-of-care-measures.csv`: Contains information about 30-day mortality and readmission rates for heart attacks, heart failure, and pneumonia for over 4,000 hospitals.
- `hospital-data.csv`: Contains information about each hospital.
- `Hospital_Revised_Flatfiles.pdf`: Descriptions of the variables in each file (codebook).

This assignment will focus on the variables for Number 19 ("Outcome of Care Measures.csv") and Number 11 ("Hospital Data.csv"). In particular, the numbers of the variables for each table indicate column indices in each table (i.e. "Hospital Name" is column 2 in the outcome-of-care-measures.csv file).

Housekeeping

First we change the working directory to be the same as where the data for this assignment is:

```
setwd("~/Desktop/data/github/Data-Science-R/Coursera/2-R-Programming")
```

Data Exploration

Read the outcome data into R

```
outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
#  read the data in as character :colClasses = "character"
head(outcome[, 1:11])
```

```
##   Provider.Number                     Hospital.Name
## 1          010001 SOUTHEAST ALABAMA MEDICAL CENTER
## 2          010005    MARSHALL MEDICAL CENTER SOUTH
## 3          010006   ELIZA COFFEE MEMORIAL HOSPITAL
## 4          010007         MIZELL MEMORIAL HOSPITAL
## 5          010008      CRENSHAW COMMUNITY HOSPITAL
## 6          010010    MARSHALL MEDICAL CENTER NORTH
##                     Address.1 Address.2 Address.3         City State
## 1     1108 ROSS CLARK CIRCLE                            DOTHAN    AL
## 2 2505 U S HIGHWAY 431 NORTH                              BOAZ    AL
## 3         205 MARENGO STREET                           FLORENCE    AL
## 4              702 N MAIN ST                                OPP    AL
## 5        101 HOSPITAL CIRCLE                            LUVERNE    AL
```

```
## 6    8000 ALABAMA HIGHWAY 69                      GUNTERSVILLE    AL
##   ZIP.Code County.Name Phone.Number
## 1    36301     HOUSTON   3347938701
## 2    35957    MARSHALL   2565938310
## 3    35631  LAUDERDALE   2567688400
## 4    36467   COVINGTON   3344933541
## 5    36049    CRENSHAW   3343353374
## 6    35976    MARSHALL   2565718000
##   Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack
## 1                                                     14.3
## 2                                                     18.5
## 3                                                     18.1
## 4                                            Not Available
## 5                                            Not Available
## 6                                            Not Available
```

```
ncol(outcome)  # number of variables
```

```
## [1] 46
```

```
nrow(outcome)  #number of observations
```

```
## [1] 4706
```

```
names(outcome)[1:11] # vector with 11 first variable names
```

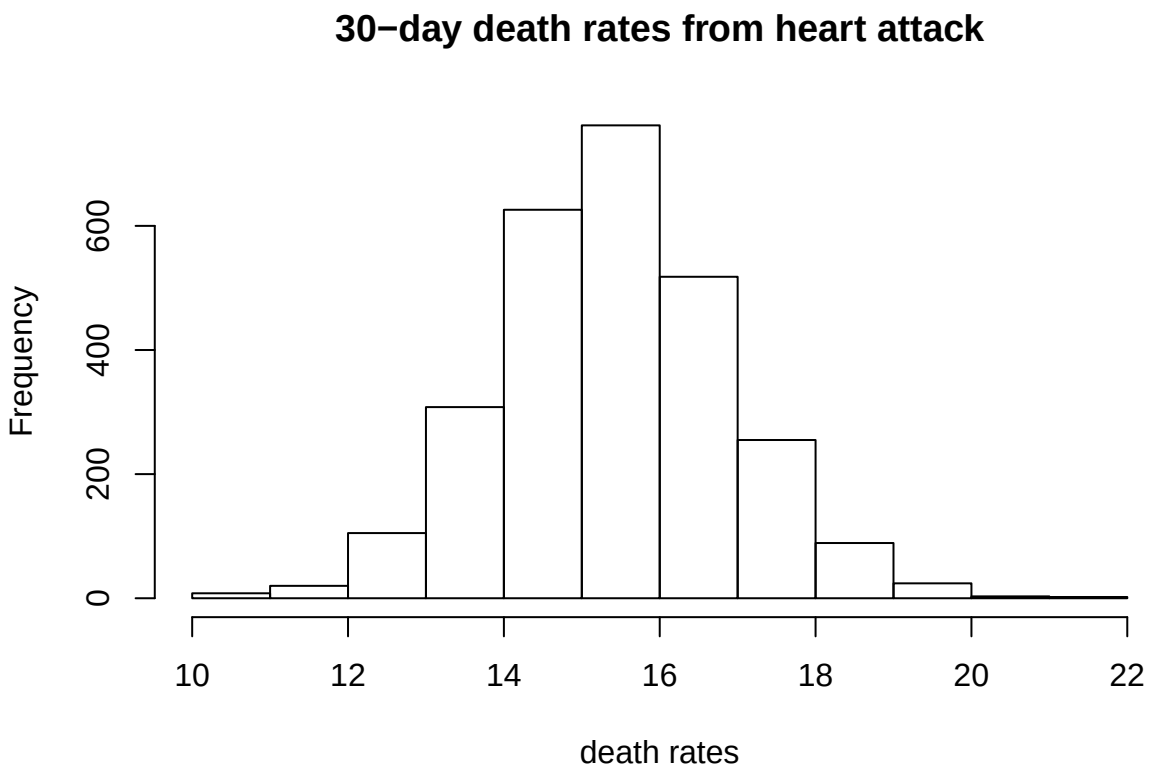
```
##  [1] "Provider.Number"
##  [2] "Hospital.Name"
##  [3] "Address.1"
##  [4] "Address.2"
##  [5] "Address.3"
##  [6] "City"
##  [7] "State"
##  [8] "ZIP.Code"
##  [9] "County.Name"
## [10] "Phone.Number"
## [11] "Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
```

Histogram of the 30-day death rates from heart attack

```
outcome[, 11] <- as.numeric(outcome[, 11]) # we need to coerce the "character" column to be numeric.
```

```
## Warning: NAs introduced by coercion
```

```
hist(outcome[, 11], main ="30-day death rates from heart attack", xlab = "death rates")
```



Exercise 1: Finding the best hospital in a state

```
best <- function(state, outcome) {

## Read outcome data
outcomes <- read.csv("outcome-of-care-measures.csv", colClasses = "character", header = TRUE)
## Keep only the variables we need for the analysis
outcomes <- outcomes[, c(2, 7, 11, 17, 23)]
## Rename them for convenience
colnames(outcomes) <- c("name", "State", "heart_attack", "heart_failure", "pneumonia")
outcome <- sub('\\ ', '_', outcome)
## Transform them to numeric
outcomes[,3]  = suppressWarnings(as.numeric(outcomes[,3]))
outcomes[,4]  = suppressWarnings(as.numeric(outcomes[,4]))
outcomes[,5]  = suppressWarnings(as.numeric(outcomes[,5]))

## Check that state and outcome are valid
if(!is.element(outcome, names(outcomes))) {
  print("invalid outcome")
} else if(!is.element(state, outcomes$State)) {
  print("invalid state")

  ## If they are, proceed with the analysis
} else {
      if(outcome == "heart_attack") {
        ##c = 3
        outcomes <- outcomes[order(outcomes$heart_attack, outcomes$name), ]
```

```
      } else if(outcome == "heart_failure") {
        ##c = 4
        outcomes <- outcomes[order(outcomes$heart_failure, outcomes$name), ]
      } else{
        ##c= 5
        outcomes <- outcomes[order(outcomes$pneumonia, outcomes$name), ]

      }

out <- outcomes[outcomes$State == state,]
out[1, 1]
}
## Another possibility would have been to use:
##outcomes <- outcomes[outcomes$State == state, ]
##out <- outcomes[, c]
## Return hospital name in that state with lowest 30-day death rate
##which(out == min(out, na.rm = TRUE)) ## Find which is the index of the minimum element

}
```

And now we test the function:

```
best("TX", "heart attack")
```

```
## [1] "CYPRESS FAIRBANKS MEDICAL CENTER"
```

```
best("TX", "heart failure")
```

```
## [1] "FORT DUNCAN MEDICAL CENTER"
```

```
best("MD", "heart attack")
```

```
## [1] "JOHNS HOPKINS HOSPITAL, THE"
```

```
best("MD", "pneumonia")
```

```
## [1] "GREATER BALTIMORE MEDICAL CENTER"
```

```
best("BB", "heart attack")
```

```
## [1] "invalid state"
```

```
best("NY", "hert attack")
```

```
## [1] "invalid outcome"
```

Exercise 2: Ranking hospitals by outcome in a state

Write a function called rankhospital that takes three arguments: the 2-character abbreviated name of a state (state), an outcome (outcome), and the ranking of a hospital in that state for that outcome (num). The function reads the outcome-of-care-measures.csv file and returns a character vector with the name of the hospital that has the ranking specified by the num argument.

```
rankhospital <- function(state, outcome, num = "best") {

##### The first part of this function is almost the same as for best() #####

## FIRST PART ## (only modifying the option na.last in order() to take out NAs)

## Read outcome data
outcomes <- read.csv("outcome-of-care-measures.csv", colClasses = "character", header = TRUE)
## Keep only the variables we need for the analysis
outcomes <- outcomes[, c(2, 7, 11, 17, 23)]
## Rename them for convenience
colnames(outcomes) <- c("name", "State", "heart_attack", "heart_failure", "pneumpnia")
outcome <- sub('\\ ', '_', outcome)
## Transform them to numeric
outcomes[,3]  = suppressWarnings(as.numeric(outcomes[,3]))
outcomes[,4]  = suppressWarnings(as.numeric(outcomes[,4]))
outcomes[,5]  = suppressWarnings(as.numeric(outcomes[,5]))

## Check that state and outcome are valid
if(!is.element(outcome, names(outcomes))) {
  stop("invalid outcome")
} else if(!is.element(state, outcomes$State)) {
  stop("invalid state")

  ## If they are, proceed with the analysis ( ++ taking out the NAs form the ranking)
} else {
      if(outcome == "heart_attack") {
        ##c = 3
        outcomes <- outcomes[order(outcomes$heart_attack, outcomes$name, na.last = NA), ]
      } else if(outcome == "heart_failure") {
        ##c = 4
        outcomes <- outcomes[order(outcomes$heart_failure, outcomes$name, na.last = NA), ]
      } else{
        ##c= 5
        outcomes <- outcomes[order(outcomes$pneumonia, outcomes$name, na.last = NA), ]

      }
  }

  out <- outcomes[outcomes$State == state, ]

  ## SECOND PART ##

  if(num == "best") {
     num = 1
  } else if(num == "worst") {
      num = length(out$name)
```

```
  }

  if(length(outcomes$name) < num){
    ans <- NA
  } else {
    ans <- out[num, 1]
  }
## Return hospital name in that state with the given rank 30-day death rate
  ans
}
```

And now we test the function:

```
rankhospital("TX", "heart failure", 4)
```

```
## [1] "DETAR HOSPITAL NAVARRO"
```

```
rankhospital("MD", "heart attack", "worst")
```

```
## [1] "HARFORD MEMORIAL HOSPITAL"
```

```
rankhospital("MN", "heart attack", 5000)
```

```
## [1] NA
```

```
rankhospital("MD", "heart failure", 5)
```

```
## [1] "SAINT AGNES HOSPITAL"
```

Exercise 3: Ranking hospitals in all states

Write a function called rankall that takes two arguments: an outcome name (outcome) and a hospital ranking (num). The function reads the outcome-of-care-measures.csv file and returns a 2-column data frame containing the hospital in each state that has the ranking specified in num.

```
rankall <- function(outcome, num = "best") {

## FIRST PART ## (same sa rankhospital, commenting out the line that subsets State == state)

## Read outcome data
outcomes <- read.csv("outcome-of-care-measures.csv", colClasses = "character", header = TRUE)
## Keep only the variables we need for the analysis
outcomes <- outcomes[, c(2, 7, 11, 17, 23)]
## Rename them for convenience
colnames(outcomes) <- c("name", "State", "heart_attack", "heart_failure", "pneumpnia")
outcome <- sub('\\ ', '_', outcome)
## Transform them to numeric
outcomes[,3]  = suppressWarnings(as.numeric(outcomes[,3]))
outcomes[,4]  = suppressWarnings(as.numeric(outcomes[,4]))
```

```
outcomes[,5]  = suppressWarnings(as.numeric(outcomes[,5]))

## Check that state and outcome are valid
if(!is.element(outcome, names(outcomes))) {
  stop("invalid outcome")
} else if(!is.element(state, outcomes$State)) {
  stop("invalid state")

  ## If they are, proceed with the analysis ( ++ taking out the NAs form the ranking)
} else {
      if(outcome == "heart_attack") {
        ##c = 3
        outcomes <- outcomes[order(outcomes$heart_attack, outcomes$name, na.last = NA), ]
      } else if(outcome == "heart_failure") {
        ##c = 4
        outcomes <- outcomes[order(outcomes$heart_failure, outcomes$name, na.last = NA), ]
      } else{
        ##c= 5
        outcomes <- outcomes[order(outcomes$pneumonia, outcomes$name, na.last = NA), ]

      }
  }

  ## out <- outcomes[outcomes$State == state, ]  ## We need all states now

  ## SECOND PART ##

final <- data.frame(hospital = sapply(State, outcome, num, rankhospital), state == State)
## Check that state and outcome are valid
## For each state, find the hospital of the given rank
## Return a data frame with the hospital names and the
## (abbreviated) state name
}
```