Programming Assignment 3: Hospital Quality: best function

Ahmed Ezzat Afifi

Thursday, October 23, 2014

Now as always, we first need to discover the story the data is telling. reading The both PDFs provided by the quiz is very important in understanding the problem we want to solve.

first lets read the data and view it

```
#Remember colClasses? just adding one means we want R to read everything as a character

outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
#You may want to use View(outcome). this won't work here on this document

#Now instead of loading the data everytime we want to access it like we did last time, we will be usi
ng a much smarter way
attach(outcome)
#This will attach the data variable to our current workspace
```

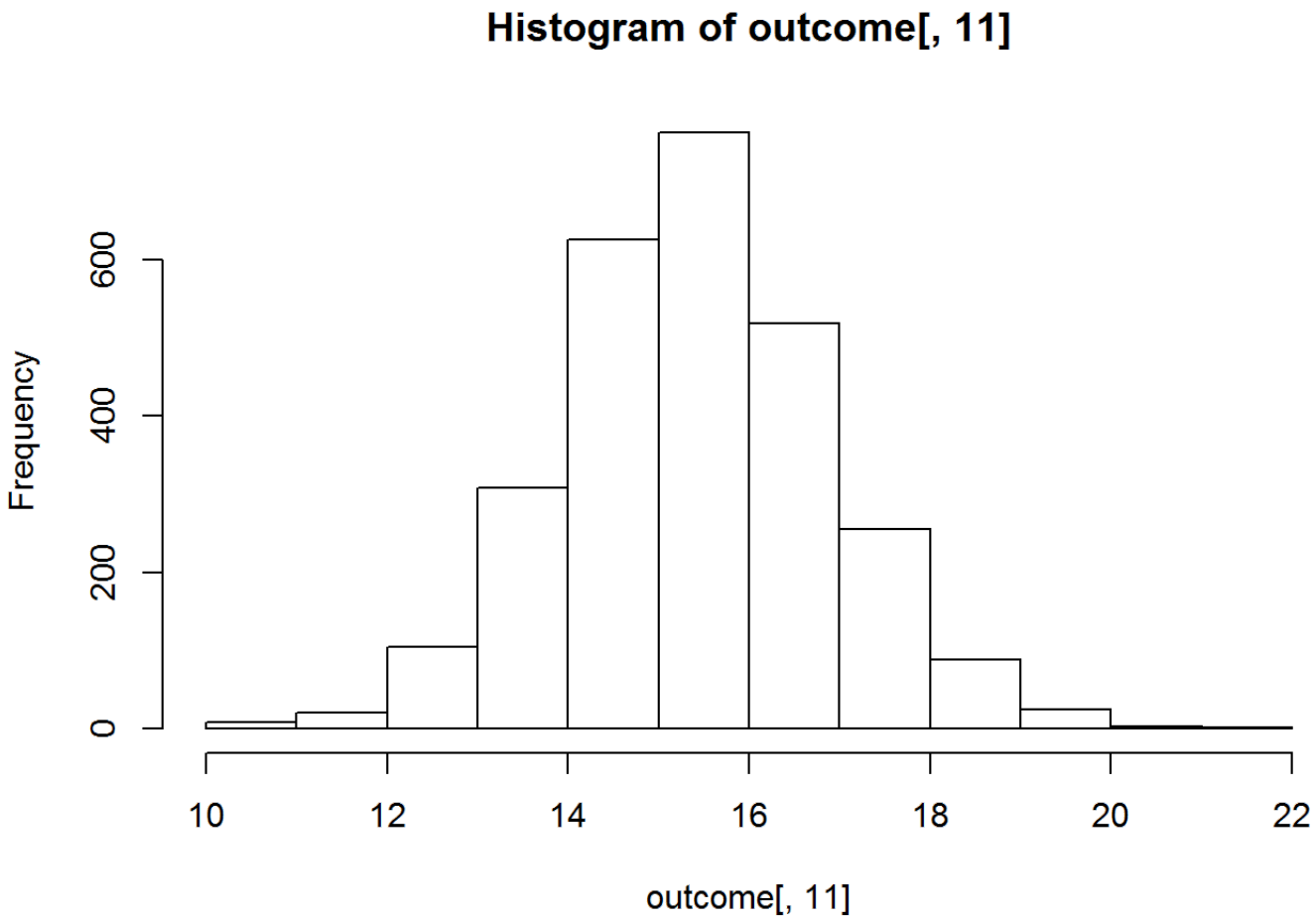
now following the PDF instructions. making a histogram for the 30-day mortality rates for heart attack

```
outcome[ ,11] <- as.numeric(outcome[ ,11]) #Change column number 11 to numeric type and then added ba
ck to the outcome data variable
```

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

```
#The NA's warning is expected since there are NA's values, and R is telling us to take care of them

hist(outcome[, 11])
```



Now read the function requirement carefully in the instructions PDF

we want to know the relevant heart failure columns in data

```
names(outcome) #Get col 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"
## [12] "Comparison.to.U.S..Rate...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
## [13] "Lower.Mortality.Estimate...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
## [14] "Upper.Mortality.Estimate...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
## [15] "Number.of.Patients...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
## [16] "Footnote...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
## [17] "Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
## [18] "Comparison.to.U.S..Rate...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
## [19] "Lower.Mortality.Estimate...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
## [20] "Upper.Mortality.Estimate...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
## [21] "Number.of.Patients...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
## [22] "Footnote...Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
## [23] "Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
## [24] "Comparison.to.U.S..Rate...Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
## [25] "Lower.Mortality.Estimate...Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
## [26] "Upper.Mortality.Estimate...Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
## [27] "Number.of.Patients...Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
## [28] "Footnote...Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
## [29] "Hospital.30.Day.Readmission.Rates.from.Heart.Attack"
## [30] "Comparison.to.U.S..Rate...Hospital.30.Day.Readmission.Rates.from.Heart.Attack"
## [31] "Lower.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Heart.Attack"
## [32] "Upper.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Heart.Attack"
## [33] "Number.of.Patients...Hospital.30.Day.Readmission.Rates.from.Heart.Attack"
## [34] "Footnote...Hospital.30.Day.Readmission.Rates.from.Heart.Attack"
## [35] "Hospital.30.Day.Readmission.Rates.from.Heart.Failure"
## [36] "Comparison.to.U.S..Rate...Hospital.30.Day.Readmission.Rates.from.Heart.Failure"
## [37] "Lower.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Heart.Failure"
## [38] "Upper.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Heart.Failure"
## [39] "Number.of.Patients...Hospital.30.Day.Readmission.Rates.from.Heart.Failure"
## [40] "Footnote...Hospital.30.Day.Readmission.Rates.from.Heart.Failure"
## [41] "Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [42] "Comparison.to.U.S..Rate...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [43] "Lower.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [44] "Upper.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [45] "Number.of.Patients...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [46] "Footnote...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
```

So the relevant columns to our function are are 11, 17 and 23

one small trick we will be using in our algorithm. consider this code

```
bands <- c("Megadeth", "Slayer", "Metalica")
myFavoriteBand <- "Slayer"
bands[bands == myFavoriteBand]
```

```
## [1] "Slayer"
```

```
#you see the Slayer in the output is not the value in myFavoriteBand variavle
#it come from bands vector
```

Developing algorithm idea

This algorithm will stay with us for the whole assignment with few modifications

1-load the data in our function

2- we have 3 possible “outcomes” possibilities for 3 columns in data. lets save those outcomes names in a vector we will call outcomevalues

3- decide which column we should use according to the relevent outcome passed in function header

4- subset the data according to the state in context in a new variable. and discard irreverent data

5- convert the relevant column to in our subset-ted data to numeric

6- remove Na’s to only story clean data

7- sort our clean data according to hospital name and pick the one with the minimum number !

so this goes like this

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

  #Loading data as they descriped in the PDF
  data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")

  #Saving the expected values of outcome argument
  outcomevalues = c("heart attack", "heart failure", "pneumonia")

  #Now a trick to select which column we need
  #We already know the indexs in the data
  #So the next line will return the number of the vector that matches the position
  #of our outcomevalues
  #So if it outcome in function header was equal to "pneumonia"
  #The next line will return 23 "POSITION number 3 in vector""

  column <- c(11, 17, 23)[outcomevalues == outcome]
```

```
  #Now we want only to save the relevent data to the passed state

  data <- data[data$State == state, ]

  #Now we want to convert the data to numeric
  #But something to take care of
  #In our exploration R throes warining when we convert to numeric
  #But the sumbmit function doesn't like R sending warnings
  #So, we will tell R to STFU about them :D
  #That's by executing what we want inside a function called suppressWarnings()
  #We just write anything we want inside it, and it will make sure R will not say anything

  #See
  suppressWarnings(data[[column]] <- as.numeric(data[[column]]))

  #Now we need to remove the missing data
  #Notice the !. it means keep everything that is not NA
  data <- data[!is.na(data[[column]]), ]

  #now we want to sort our data, here the function order() comes to help
  #Read ?order
  #using it we can sort data by name too. we can do that with sort() function
  #The best is the first row
  data <- data[order(data[column], data$Hospital.Name), ]

  #And that's it
  #We should return the first name of a hospital in our ordered data

  data$Hospital.Name[1]

}
```

Now save that function in a R script called best.R and submit it