Hospital Data

Matthew Dunne

Assignment

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. The purpose of the web site is to provide data and information about the quality of care at over 4,000 Medicare-certi???ed hospitals in the U.S. This dataset essentially covers all major U.S. hospitals.

1. Plot the 30-day mortality rates for heart attack

Read the outcome data into R via the read.csv function and look at the ???rst few rows.

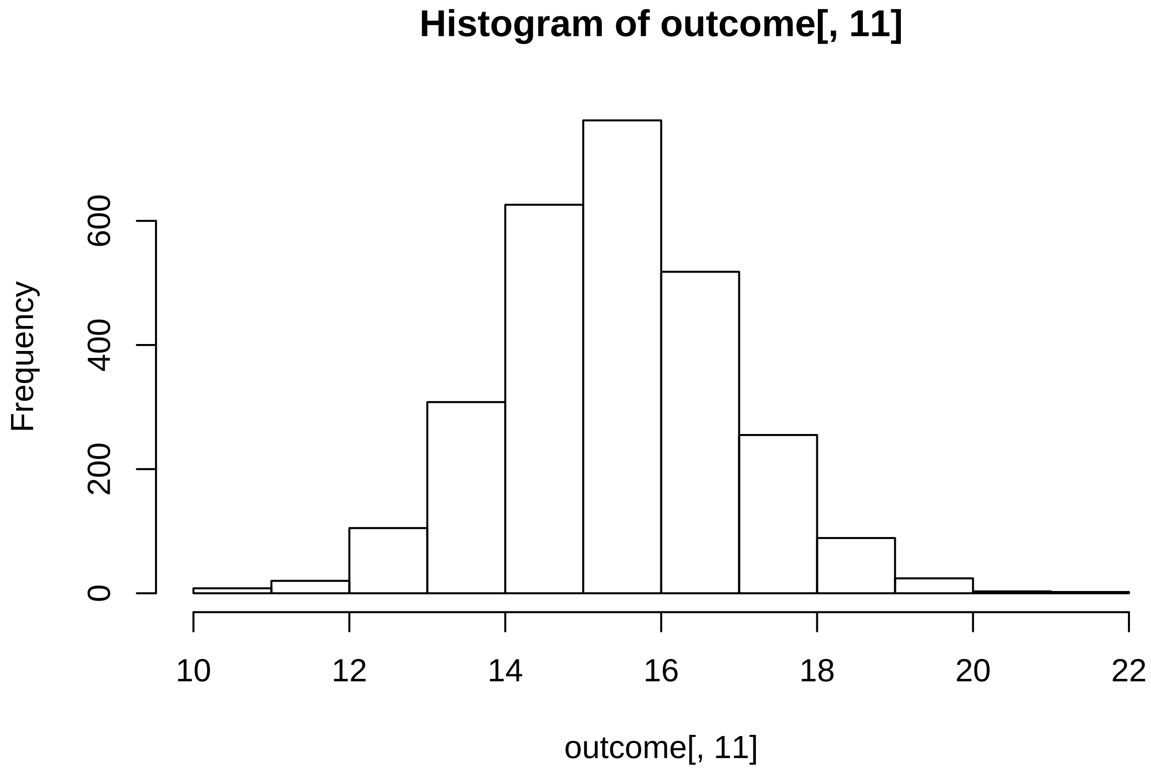
```
outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
#head(outcome)
```

Make a simple histogram of the 30-day death rates from heart attack (column 11 in the outcome dataset).

```
outcome[, 11] <- as.numeric(outcome[, 11])
```

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

```
hist(outcome[, 11])
```



2 Finding the best hospital in a state

Write a function called best that take two arguments: the 2-character abbreviated name of a state and an outcome name. The function reads the outcome-of-care-measures.csv ???le and returns a character vector with the name of the hospital that has the best (i.e. lowest) 30-day mortality for the speci???ed outcome in that state. The hospital name is the name provided in the Hospital.Name variable.

The outcomes can be one of “heart attack”, “heart failure”, or “pneumonia”. Hospitals that do not have data on a particular outcome should be excluded from the set of hospitals when deciding the rankings.

Handling ties. If there is a tie for the best hospital for a given outcome, then the hospital names should be sorted in alphabetical order and the ???rst hospital in that set should be chosen (i.e. if hospitals “b”, “c”, and “f” are tied for best, then hospital “b” should be returned).

```
best<-function(state=NULL, outcome=NULL){
  setwd(file.path("C:/Users/mjdun/Desktop/Coursera/Data Science Specialization/Course 2 R Programming/W
  data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
  ##create list of outcomes to narrow down data
  outcomes<-list("heart attack", "heart failure", "pneumonia")
  ## check if state and outcome are valid entries
  if(!(state %in% data[ ,7]))
      stop("invalid state")
  if(!(outcome %in% outcomes))
     stop("invalid outcome")
  ##narrow data down to appropriate state (go by row hence the last comma)
  data<-data[data$State==state, ]
  ##narrow data down to appropriate outcome mortality rates
  if(outcome==outcomes[1]){
    data<-data[ ,c(2, 7, 11)]
    }
  if(outcome==outcomes[2]){
    data<-data[ ,c(2, 7, 17)]
    }
  if(outcome==outcomes[3]){
    data<-data[ ,c(2, 7, 23)]
    }
  ##convert mortality rate to numeric so you can compare it
  data[ ,3] <-as.numeric(data[,3])
  ##take out NAs generated by converting to numeric
  data<-na.omit(data)
  ##order by mortality rate then hospital name (go by row hence the last comma)
  data<-data[with(data, order(data[, 3], data[ ,1])), ]
  ##print only the hospital name (col 1) of the first row of the ordered set
  print(data[1, 1])
}
```

The function should check the validity of its arguments. If an invalid state value is passed to best, the function should throw an error via the stop function with the exact message “invalid state”. If an invalid outcome value is passed to best, the function should throw an error via the stop function with the exact message “invalid outcome”.

Here is some sample output from the function.

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

```
## Warning in best("TX", "heart attack"): NAs introduced by coercion
## [1] "CYPRESS FAIRBANKS MEDICAL CENTER"
```

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

```
## Warning in best("TX", "heart failure"): NAs introduced by coercion
```

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

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

```
## Warning in best("MD", "heart attack"): NAs introduced by coercion
```

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

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

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

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

```
## Warning in best("TX", "heart attack"): NAs introduced by coercion
```

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

3. 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 ???le and returns a character vector with the name of the hospital that has the ranking speci???ed by the num argument.

For example, the call

rankhospital(“MD”, “heart failure”, 5)

would return a character vector containing the name of the hospital with the 5th lowest 30-day death rate for heart failure. The num argument can take values “best”, “worst”, or an integer indicating the ranking (smaller numbers are better). If the number given by num is larger than the number of hospitals in that state, then the function should return NA. Hospitals that do not have data on a particular outcome should be excluded from the set of hospitals when deciding the rankings.

Handling ties. It may occur that multiple hospitals have the same 30-day mortality rate for a given cause of death. In those cases ties should be broken by using the hospital name.

The function should check the validity of its arguments. If an invalid state value is passed to rankhospital, the function should throw an error via the stop function with the exact message“invalid state”. If an invalid outcome value is passed to rankhospital, the function should throw an error via the stop function with the exact message “invalid outcome”.

```r
rankhospital<-function(state=NULL, outcome=NULL, num="best"){
  setwd(file.path("C:/Users/mjdun/Desktop/Coursera/Data Science Specialization/Course 2 R Programming/We
  data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
  ##create list of outcomes to narrow down data
  outcomes<-list("heart attack", "heart failure", "pneumonia")
  ## check if state and outcome are valid entries
  if(!(state %in% data[ ,7]))
    stop("invalid state")
  if(!(outcome %in% outcomes))
    stop("invalid outcome")
  ##narrow data down to appropriate state (go by row hence the last comma)
  data<-data[data$State==state, ]
  ##narrow data down to appropriate outcome mortality rates
  if(outcome==outcomes[1]){
    data<-data[ ,c(2, 7, 11)]
  }
  if(outcome==outcomes[2]){
    data<-data[ ,c(2, 7, 17)]
  }
  if(outcome==outcomes[3]){
    data<-data[ ,c(2, 7, 23)]
  }
  ##convert mortality rate to numeric so you can compare it
  data[ ,3] <-as.numeric(data[,3])
  ##take out NAs
  data<-na.omit(data)
  ##order by mortality rate then hospital name (go by row hence the last comma)
  data<-data[with(data, order(data[, 3], data[ ,1])), ]
  ##new variable = how many records there are
  records<-nrow(data)
  ##decide which row to use based on whether best, worst, too large a number, or other
  if (num=="best"){
    data<-data[1,1]}
  else if (num=="worst"){
    data<-tail(data, n=1)
    data<-data[1,1]}
  else if (num>records){
    return(NA)
  }
  else if(!(num=="best" & num=="worst")){
    ##make sure it is an integer so you can plug it in
    num<-as.integer(num)
    data<-data[num,1]
  }
  print(data)
}
```

Here is some sample output from the function.

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

```
## Warning in rankhospital("TX", "heart failure", 4): NAs introduced by
## coercion

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

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

```
## Warning in rankhospital("MD", "heart attack", "worst"): NAs introduced by
## coercion

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

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

```
## Warning in rankhospital("MN", "heart attack", 5000): NAs introduced by
## coercion
## [1] NA
```

4. 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 ???le and returns a 2-column data frame containing the hospital in each state that has the ranking speci???ed in num. For example the function call rankall(“heart attack”, “best”) would return a data frame containing the names of the hospitals that are the best in their respective states for 30-day heart attack death rates. The function should return a value for every state (some may be NA). The ???rst column in the data frame is named hospital, which contains the hospital name, and the second column is named state, which contains the 2-character abbreviation for the state name. Hospitals that do not have data on a particular outcome should be excluded from the set of hospitals when deciding the rankings.

Handling ties. The rankall function should handle ties in the 30-day mortality rates in the same way that the rankhospital function handles ties.

The function should check the validity of its arguments. If an invalid outcome value is passed to rankall, the function should throw an error via the stop function with the exact message“invalid outcome”. The num variable can take values “best”, “worst”, or an integer indicating the ranking (smaller numbers are better). If the number given by num is larger than the number of hospitals in that state, then the function should return NA.

```r
rankall<-function(outcome=NULL, num="best"){
  setwd(file.path("C:/Users/mjdun/Desktop/Coursera/Data Science Specialization/Course 2 R Programming/W
  data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
  ##create character vector of list of all states in data
  states<-unique(data$State)
  ##create list of outcomes to narrow down data
  outcomes<-list("heart attack", "heart failure", "pneumonia")
  ## check if state and outcome are valid entries
  if(!(outcome %in% outcomes))
    stop("invalid outcome")
  list<-list()
  masterdata<-data.frame()
  ##narrow data down to appropriate state (go by row hence the last comma)
  for (i in states){
    state<-i
    newdata<-subset(data, data$State==state)
    ##new variable = how many records there are
    records<-nrow(newdata)
      ##narrow data down to appropriate outcome mortality rates
      if(outcome==outcomes[1]){
        newdata<-newdata[ ,c(2, 7, 11)]}
      if(outcome==outcomes[2]){
        newdata<-newdata[ ,c(2, 7, 17)]}
      if(outcome==outcomes[3]){
        newdata<-newdata[ ,c(2, 7, 23)]}
      ##convert mortality rate to numeric so you can compare it
    newdata[ ,3] <-as.numeric(newdata[,3])
      ##take out NAs
    newdata<-na.omit(newdata)
      ##order by mortality rate then hospital name (go by row hence the last comma)
    newdata<-newdata[with(newdata, order(newdata[, 3], newdata[ ,1])), ]
```

```
      ##decide which row to use based on whether best, worst, too large a number, or other
    if (num=="best"){
        newdata<-newdata[1,1]}
    else if (num=="worst"){
        newdata<-tail(newdata, n=1)
        newdata<-newdata[1,1]}
    ##else if (num>records){ (as written returns NA for the whole thing. Without just returns NA for Gua
        ##return(NA)}
    else if(!(num=="best" & num=="worst")){
        ##make sure it is an integer so you can plug it in
        num<-as.integer(num)
        newdata<-newdata[num,1]}
    x<-data.frame(hospital=newdata, state=state)
      masterdata<-rbind(masterdata, x)
  }
  suppressWarnings(masterdata)
}
```

Here is some sample output from the function.

```
head(suppressWarnings(rankall("heart attack", 20)), 10)
```

```
##                                hospital state
## 1       D W MCMILLAN MEMORIAL HOSPITAL    AL
## 2                                 <NA>    AK
## 3  JOHN C LINCOLN DEER VALLEY HOSPITAL    AZ
## 4    ARKANSAS METHODIST MEDICAL CENTER    AR
## 5                SHERMAN OAKS HOSPITAL    CA
## 6             SKY RIDGE MEDICAL CENTER    CO
## 7              MIDSTATE MEDICAL CENTER    CT
## 8                                 <NA>    DE
## 9                                 <NA>    DC
## 10      SOUTH FLORIDA BAPTIST HOSPITAL    FL
```

```
tail(suppressWarnings(rankall("pneumonia", "worst")), 3)
```

```
##                                       hospital state
## 52 MAYO CLINIC HEALTH SYSTEM - NORTHLAND, INC    WI
## 53           NORTH BIG HORN HOSPITAL DISTRICT    WY
## 54           GUAM MEMORIAL HOSPITAL AUTHORITY    GU
```

```
tail(suppressWarnings(rankall("heart failure")), 10)
```

```
##                                                              hospital state
## 45                                        FORT DUNCAN MEDICAL CENTER    TX
## 46 VA SALT LAKE CITY HEALTHCARE - GEORGE E. WAHLEN VA MEDICAL CENTER    UT
## 47                                              SPRINGFIELD HOSPITAL    VT
## 48                            GOV JUAN F LUIS HOSPITAL & MEDICAL CTR    VI
## 49                                          SENTARA POTOMAC HOSPITAL    VA
## 50                                         HARBORVIEW MEDICAL CENTER    WA
## 51                                         FAIRMONT GENERAL HOSPITAL    WV
## 52                                    AURORA ST LUKES MEDICAL CENTER    WI
## 53                                        CHEYENNE VA MEDICAL CENTER    WY
## 54                                  GUAM MEMORIAL HOSPITAL AUTHORITY    GU
```