R Notebook

Matteo Gambera

2020-02-29

1 Plot the 30-day mortality rates for heart attack

I import the dataset with related data related to hospital and disease

```
outcome <- read.csv("/home/matteo/Scrivania/datasciencecoursera/Course_2/Data/hospital_compare/outcome-
```

the dataset is quite large, I check the number of columns and rows.

```
ncol(outcome)
```

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

```
nrow(outcome)
```

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

I therefore have 4706 hospitals with 46 information on it. I can also see the information I have about patients.

```
columns_names <- names(outcome)
head(columns_names)
```

```
## [1] "Provider.Number" "Hospital.Name"   "Address.1"       "Address.2"
## [5] "Address.3"       "City"
```

```
tail(columns_names)
```

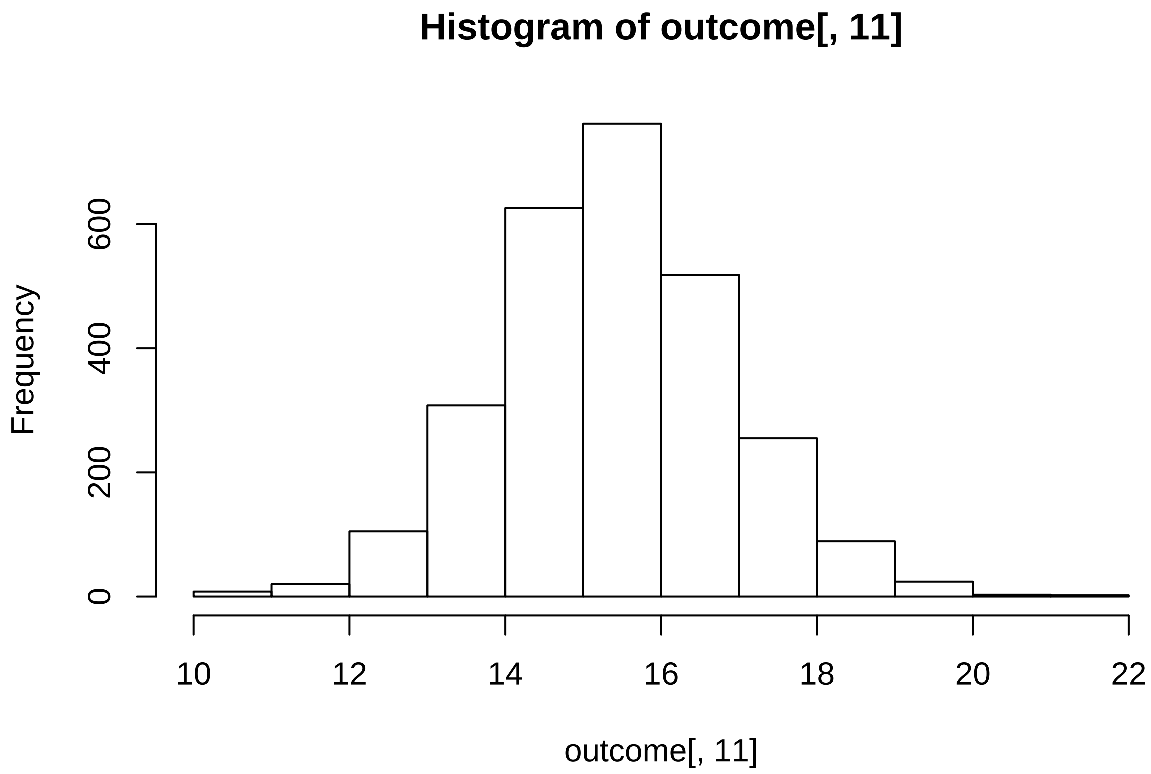
```
## [1] "Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [2] "Comparison.to.U.S..Rate...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [3] "Lower.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [4] "Upper.Readmission.Estimate...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [5] "Number.of.Patients...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
## [6] "Footnote...Hospital.30.Day.Readmission.Rates.from.Pneumonia"
```

I create a small histogram to look at heart attack deaths in the past 30 days. The column associated with this data is 11.

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

```
## Warning: si è prodotto un NA per coercizione
```

```r
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 file and returns a character vector with the name of the hospital that has the best (i.e. lowest) 30-day mortality for the specified 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.

```r
hospitals <- (outcome$Hospital.Name)
head(hospitals)
```

```
## [1] "SOUTHEAST ALABAMA MEDICAL CENTER" "MARSHALL MEDICAL CENTER SOUTH"
## [3] "ELIZA COFFEE MEMORIAL HOSPITAL"   "MIZELL MEMORIAL HOSPITAL"
## [5] "CRENSHAW COMMUNITY HOSPITAL"      "MARSHALL MEDICAL CENTER NORTH"
```

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".

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

     ## All type
    outcomes = c("heart attack", "heart failure", "pneumonia")

    ## Read outcome data
    data <- read.csv("/home/matteo/Scrivania/datasciencecoursera/Course_2/Data/hospital_compare/outcome-

    ## Check that state and outcome are valid
    ## build a new table structured like this  ("name", "state", "heart attack",...)
    #"Hospital.Name"
    #"State"
    #"Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack"
    #"Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure"
    #"Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia"
    data <- data[c(2,7,11,17,23)]
    ## rename headers
    names(data)[1] <- "name"
    names(data)[2] <- "state"
    names(data)[3] <- "heart attack"
    names(data)[4] <- "heart failure"
    names(data)[5] <- "pneumonia"

    ## Check for stases and outcome
    states <- data[,2]
    states <- unique(states) # unique returns a vector, data frame or array like x but with duplicate e
    if( state %in% states == FALSE ) {
        stop("invalid state")
    }
    if(outcome!= "heart attack" & outcome != "heart failure" & outcome != "pneumonia")
    {

        stop("invalid outcome")
    }
    data <- data[data$state==state & data[outcome] != 'Not Available', ]



    ## Return hospital name in that state with lowest 30-day death
    ## rate
    values <- data[, outcome]
    min_death_hospital <- which.min(values)
    data[min_death_hospital, ]$name

}
```

Test for best 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("NY", "heart attack")
```

```
## [1] "NYU HOSPITALS CENTER"
```

```
#best("BB", "heart attack") return error for BB state
```

```
#best("NY", "hert attack") return errpr for hert attack
best("SC", "heart attack")
```

```
## [1] "MUSC MEDICAL CENTER"
```

```
best("NY", "pneumonia")
```

```
## [1] "MAIMONIDES MEDICAL CENTER"
```

```
best("AK", "pneumonia")
```

```
## [1] "YUKON KUSKOKWIM DELTA REG HOSPITAL"
```

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 file and returns a character vector with the name of the hospital that has the ranking specified 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.

```r
rankhospital <- function(state, outcome, num) {

    ## All of the possible outcome strings
    outcomes = c("heart attack", "heart failure", "pneumonia")

    ## Read outcome data .csv file
    data <- read.csv("/home/matteo/Scrivania/datasciencecoursera/Course_2/Data/hospital_compare/outcome-
    data <- data[c(2, 7, 11, 17, 23)]
    names(data)[1] <- "name"
    names(data)[2] <- "state"
    names(data)[3] <- "heart attack"
    names(data)[4] <- "heart failure"
    names(data)[5] <- "pneumonia"


    ## Check if outcome is one of the strings in outcomes
    if( outcome %in% outcomes == FALSE ) {
        stop("invalid outcome")
    }

    ## Validate the state string
    ## All of the possible states from the data
    states <- data[, 2]
    states <- unique(states)
    ## Check if state is one of the states in the data
    if( state %in% states == FALSE ) {
        stop("invalid state")
    }

    ## Validate the num value by checking if it is "best","worst", or a number.
    ## NOTE: If num was Boolean then num%%1 = 0
    if( num != "best" && num != "worst" && num%%1 != 0 ) {
        stop("invalid num")
    }

    ## Get only the rows with our state value
    data <- data[data$state==state & data[outcome] != 'Not Available', ]

    ## Order the data by name and then outcome
    data[outcome] <- as.data.frame(sapply(data[outcome], as.numeric))
    data <- data[order(data$name, decreasing = FALSE), ]
    data <- data[order(data[outcome], decreasing = FALSE), ]

    ## Process the num argument to get the row index
    values <- data[, outcome]
    if( num == "best" ) {
        rowNum <- which.min(values)
    } else if( num == "worst" ) {
        rowNum <- which.max(values)
    } else {
        rowNum <- num
    }
    ## Return hospital name in that state with lowest 30-day death rate
    data[rowNum, ]$name
}
```

Test for rankhospital function

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

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

```
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("NC", "heart attack", "worst")
```

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

```
rankhospital("WA", "heart attack", 7)
```

```
## [1] "YAKIMA VALLEY MEMORIAL HOSPITAL"
```

```
rankhospital("TX", "pneumonia", 10)
```

```
## [1] "SETON SMITHVILLE REGIONAL HOSPITAL"
```

```
rankhospital("NY", "heart attack", 7)
```

```
## [1] "BELLEVUE HOSPITAL CENTER"
```

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 file and returns a 2-column data frame containing the hospital in each state that has the ranking specified 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 first 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.

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

    ## Read outcome data
    data <- read.csv("/home/matteo/Scrivania/datasciencecoursera/Course_2/Data/hospital_compare/outcome-
    data <- data[c(2, 7, 11, 17, 23)]
    names(data)[1] <- "name"
    names(data)[2] <- "state"
    names(data)[3] <- "heart attack"
    names(data)[4] <- "heart failure"
    names(data)[5] <- "pneumonia"

    ## Validate the outcome string
    outcomes = c("heart attack", "heart failure", "pneumonia")
    if( outcome %in% outcomes == FALSE ) stop("invalid outcome")

    ## Validate the num value
    if( num != "best" && num != "worst" && num%%1 != 0 ) stop("invalid num")

    ## Grab only rows with data in our outcome
    data <- data[data[outcome] != 'Not Available', ]

    ## Order the data
    data[outcome] <- as.data.frame(sapply(data[outcome], as.numeric))
    data <- data[order(data$name, decreasing = FALSE), ]
    data <- data[order(data[outcome], decreasing = FALSE), ]

    ## Helper functiont to process the num argument
    getHospByRank <- function(df, s, n) {
        df <- df[df$state==s, ]
        vals <- df[, outcome]
        if( n == "best" ) {
            rowNum <- which.min(vals)
        } else if( n == "worst" ) {
            rowNum <- which.max(vals)
        } else {
            rowNum <- n
        }
        df[rowNum, ]$name
    }

    ## For each state, find the hospital of the given rank
    states <- data[, 2]
    states <- unique(states)
    newdata <- data.frame("hospital"=character(), "state"=character())
    for(st in states) {
        hosp <- getHospByRank(data, st, num)
        newdata <- rbind(newdata, data.frame(hospital=hosp, state=st))
    }

    ## Return a data frame with the hospital names and the (abbreviated) state name
    newdata <- newdata[order(newdata['state'], decreasing = FALSE), ]
    newdata
}
```

Test for rankall function

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

```
##                               hospital state
## 42                                <NA>    AK
## 40      D W MCMILLAN MEMORIAL HOSPITAL    AL
## 11   ARKANSAS METHODIST MEDICAL CENTER    AR
## 16 JOHN C LINCOLN DEER VALLEY HOSPITAL    AZ
## 4                SHERMAN OAKS HOSPITAL    CA
## 24            SKY RIDGE MEDICAL CENTER    CO
## 5              MIDSTATE MEDICAL CENTER    CT
## 49                                <NA>    DC
## 31                                <NA>    DE
## 19      SOUTH FLORIDA BAPTIST HOSPITAL    FL
```

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

```
##                                      hospital state
## 28 MAYO CLINIC HEALTH SYSTEM - NORTHLAND, INC    WI
## 36                     PLATEAU MEDICAL CENTER    WV
## 41           NORTH BIG HORN HOSPITAL DISTRICT    WY
```

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

```
##                                                              hospital state
## 36                         WELLMONT HAWKINS COUNTY MEMORIAL HOSPITAL    TN
## 12                                        FORT DUNCAN MEDICAL CENTER    TX
## 50 VA SALT LAKE CITY HEALTHCARE - GEORGE E. WAHLEN VA MEDICAL CENTER    UT
## 15                                          SENTARA POTOMAC HOSPITAL    VA
## 54                            GOV JUAN F LUIS HOSPITAL & MEDICAL CTR    VI
## 52                                              SPRINGFIELD HOSPITAL    VT
## 23                                         HARBORVIEW MEDICAL CENTER    WA
## 31                                    AURORA ST LUKES MEDICAL CENTER    WI
## 38                                         FAIRMONT GENERAL HOSPITAL    WV
## 47                                        CHEYENNE VA MEDICAL CENTER    WY
```

```r
r <- rankall("heart attack", 4)
as.character(subset(r, state == "HI")$hospital)
```

```
## [1] "CASTLE MEDICAL CENTER"
```

```r
r <- rankall("pneumonia", "worst")
as.character(subset(r, state == "NJ")$hospital)
```

```
## [1] "BERGEN REGIONAL MEDICAL CENTER"
```

```r
r <- rankall("heart failure", 10)
as.character(subset(r, state == "NV")$hospital)
```

```
## [1] "RENOWN SOUTH MEADOWS MEDICAL CENTER"
```