Programming assignment 3 - Week 4

Raphael Carvalho

21/06/2019

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 first few rows.

```
outcome <- read.csv("./rprog-data-ProgAssignment3-data/outcome-of-care-measures.csv",colClasses = "chara
#head(outcome)
```

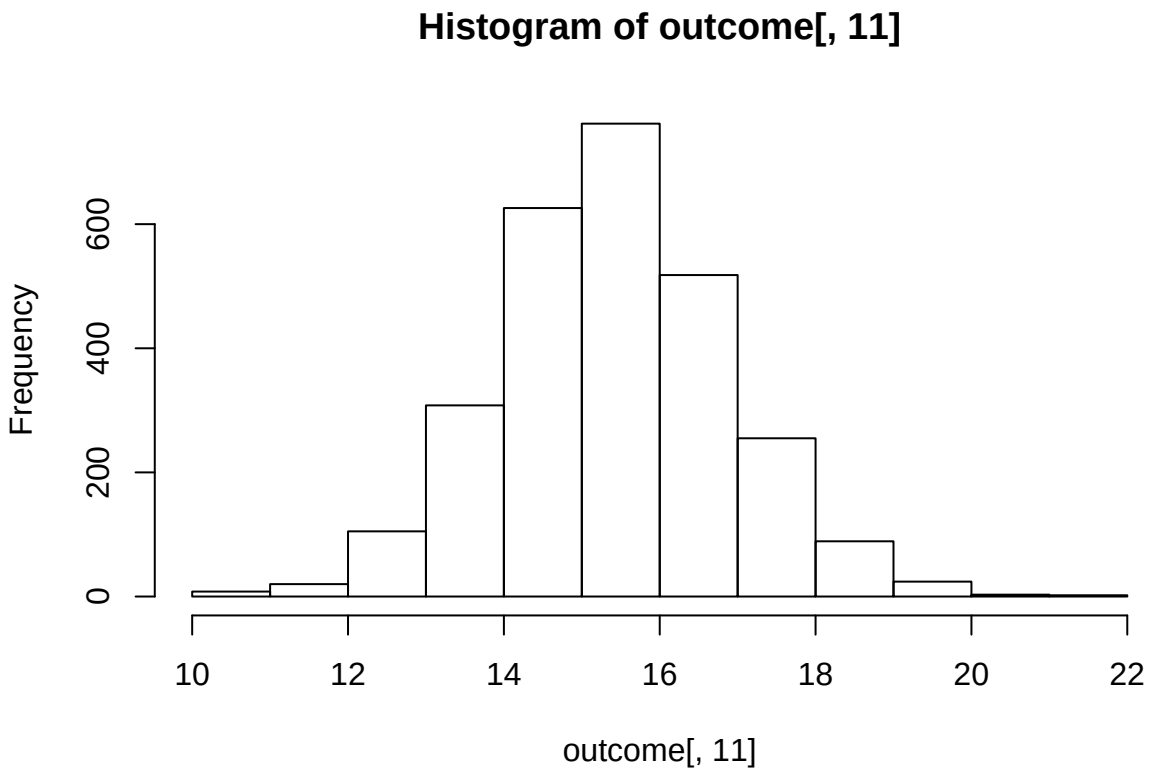
There are many columns in this dataset. You can see how many by typing ncol(outcome) (you can see the number of rows with the nrow function). In addition, you can see the names of each column by typing names(outcome) (the names are also in the PDF document.

To make a simple histogram of the 30-day death rates from heart attack (column 11 in the outcome dataset), run

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

```
## Warning: NAs introduzidos por coerção
```

```
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.

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 first 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, outcome) {

  ## Read outcome data
  dt <- read.csv("./rprog-data-ProgAssignment3-data/outcome-of-care-measures.csv", colClasses = "charac

  ## Check that state and outome are valid
  outcomes <- c("heart failure", "heart attack", "pneumonia")
  states <- as.character(unique(dt[, "State"]))
  if (!(state %in% states)) {
    return(stop("invalid state"))
  }
  if (!(outcome %in% outcomes)) {
    return(stop("invalid outcome"))
  }

  ## Return hospital name in that state with lowest 30-day death rate

  colnames.norm <- gsub("\\.", " ", colnames(dt))
  metrics.cols <- colnames.norm[grep("^Hospital 30 Day Death  Mortality  Rates from", colnames.norm)]
  match.col <- metrics.cols[grep(outcome, metrics.cols, ignore.case = TRUE)]
  colnames(dt) <- colnames.norm

  # Subset dataset columns
  dt <- dt[, c("State", "Hospital Name", match.col)]
  dt <- dt[which(dt$State == state), c("Hospital Name", match.col)]
  dt[[match.col]] <- suppressWarnings(as.numeric(dt[[match.col]], na.rm = TRUE))

  # Get the hospital name
  hospitals <- dt[which(dt[[match.col]] == min(dt[[match.col]], na.rm = TRUE)), "Hospital Name"]
  hospitals <- sort(hospitals, decreasing = FALSE)
  hospitals[1]
}
```

Tests

```
ifelse(best("TX", "heart attack") == "CYPRESS FAIRBANKS MEDICAL CENTER", TRUE, FALSE)
```

```
## [1] TRUE
```

```
ifelse(best("TX", "heart failure") == "FORT DUNCAN MEDICAL CENTER", TRUE, FALSE)
```

```
## [1] TRUE
```

```
ifelse(best("MD", "heart attack") == "JOHNS HOPKINS HOSPITAL, THE", TRUE, FALSE)
```

```
## [1] TRUE
```

```
ifelse(best("MD", "pneumonia") == "GREATER BALTIMORE MEDICAL CENTER", TRUE, FALSE)
```

```
## [1] TRUE
```

```
ifelse(tryCatch(best("BB", "heart attack"), finally = print(TRUE)) == TRUE, TRUE, FALSE)
```

```
## Error in best("BB", "heart attack"): invalid state
```

```
## [1] TRUE
```

```
ifelse(tryCatch(best("NY", "hert attack"), finally = print(TRUE)) == TRUE, TRUE, FALSE)
```

```
## Error in best("NY", "hert attack"): invalid outcome
```

```
## [1] TRUE
```

[1] "JOHNS HOPKINS HOSPITAL, THE" > best("MD", "pneumonia")

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.

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. For example, in Texas ("TX"), the hospitals with lowest 30-day mortality rate for heart failure are shown here.

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

  ## Read outcome data
  dt <- read.csv("./rprog-data-ProgAssignment3-data/outcome-of-care-measures.csv", colClasses = "charac

  ## Check that state and outome are valid
  outcomes <- c("heart failure", "heart attack", "pneumonia")
  states <- as.character(unique(dt[, "State"]))
  if (!(state %in% states)) {
    return(stop("invalid state"))
  }
  if (!(outcome %in% outcomes)) {
    return(stop("invalid outcome"))
  }


  ## Return hospital name in that state with the required rank for 30-day death rate

  colnames.norm <- gsub("\\.", " ", colnames(dt))
  metrics.cols <- colnames.norm[grep("^Hospital 30 Day Death  Mortality  Rates from", colnames.norm)]
  match.col <- metrics.cols[grep(outcome, metrics.cols, ignore.case = TRUE)]
  colnames(dt) <- colnames.norm

  # Subset dataset columns
  dt <- dt[, c("State", "Hospital Name", match.col)]
  dt <- dt[which(dt$State == state), c("Hospital Name", match.col)]
```

```
  dt[[match.col]] <- suppressWarnings(as.numeric(dt[[match.col]], na.rm = TRUE))
  dt <- dt[order(dt[[match.col]]),]
  if (num %in% c("best", "worst")) {
    if (num == "best") {
      dt <- dt[order(dt[[match.col]], dt[["Hospital Name"]]),]
      rank <- dt[1, "Hospital Name"]
    } else {
      dt <- dt[order(dt[[match.col]], dt[["Hospital Name"]], decreasing = TRUE),]
      rank <- dt[1, "Hospital Name"]
    }
  } else {
    dt <- dt[order(dt[[match.col]], dt[["Hospital Name"]]),]
    rank <- dt[num, "Hospital Name"]
  }
  rank
}
```

Testing

```
ifelse(rankhospital("TX", "heart failure", 4) == "DETAR HOSPITAL NAVARRO", TRUE, FALSE)
```

```
## [1] TRUE
```

```
ifelse(rankhospital("MD", "heart attack", "worst") == "HARFORD MEMORIAL HOSPITAL", TRUE, FALSE)
```

```
## [1] TRUE
```

```
ifelse(is.na(rankhospital("MN", "heart attack", 5000)) == TRUE, TRUE, FALSE)
```

```
## [1] TRUE
```

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.

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

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

  ## Read outcome data
  dt <- read.csv("./rprog-data-ProgAssignment3-data/outcome-of-care-measures.csv", colClasses = "charac

  ## Check that state and outome are valid
  outcomes <- c("heart failure", "heart attack", "pneumonia")
  if (!(outcome %in% outcomes)) {
    return(stop("invalid outcome"))
  }
```

```
  ## Normalizing column names

  colnames.norm <- gsub("\\.", " ", colnames(dt))
  metrics.cols <- colnames.norm[grep("^Hospital 30 Day Death  Mortality  Rates from", colnames.norm)]
  match.col <- metrics.cols[grep(outcome, metrics.cols, ignore.case = TRUE)]
  colnames(dt) <- colnames.norm

  ## Subsetting the dataset
  dt <- dt[, c("Hospital Name", "State", match.col)]
  colnames(dt) <- c("hospital", "state", "metric")
  dt$metric <- as.numeric(dt$metric)
  dt <- dt[!is.na(dt$metric), ]

  ## Creating a list of dataframes for each state
   splitted.dt <- split(dt, dt$state)
   rank <- lapply(splitted.dt, function(x, num) {
     x <- x[order(x$metric, x$hospital), ]
     if (num == "best") {
       return (x$hospital[1])
     } else if (num == "worst") {
       return(x$hospital[nrow(x)])
     } else {
       return(x$hospital[num])
     }
   },num)
  return(data.frame(hospital=unlist(rank), state=names(rank)))
}
```

Testing

```
first <- c('D W MCMILLAN MEMORIAL HOSPITAL', 'SOUTH FLORIDA BAPTIST HOSPITAL')
ifelse(head(rankall("heart attack", 20), 10)$hospital[c(2, 10)] == first, TRUE, FALSE)
```

```
## [1] TRUE TRUE
```

```
second <- c("MAYO CLINIC HEALTH SYSTEM - NORTHLAND, INC", "PLATEAU MEDICAL CENTER", "NORTH BIG HORN HOSP
ifelse(tail(rankall("pneumonia", "worst"), 3)$hospital == second, TRUE, FALSE)
```

```
## [1] TRUE TRUE TRUE
```

```
third <- c("WELLMONT HAWKINS COUNTY MEMORIAL HOSPITAL", "SENTARA POTOMAC HOSPITAL")
ifelse(tail(rankall("heart failure"), 10)$hospital[c(1, 4)] == third, TRUE, FALSE)
```

```
## [1] TRUE TRUE
```