PR programming Assignment 3 week 4

Gustavo Ahumada

12/19/2020

Programming Assignment 3

Course 2 R Programming, Assignment 3 (Week 4), under Data Science by Johns Hopkins University

To begin, download data and unzip it into your R working directory.

You can do this in R with the following code:

```
knitr::opts_knit$set(root.dir = '/Users/gusahu/Google Drive/Online_courses/r_programming/rprogramming_c
getwd()
```

```
## [1] "/Users/gusahu/Google Drive/Online_courses/r_programming/rprogramming_coursera/week_4"
```

```
dataset <- getwd()
unzip("rprog_data_ProgAssignment3-data.zip", exdir = "hosp_compare")
```

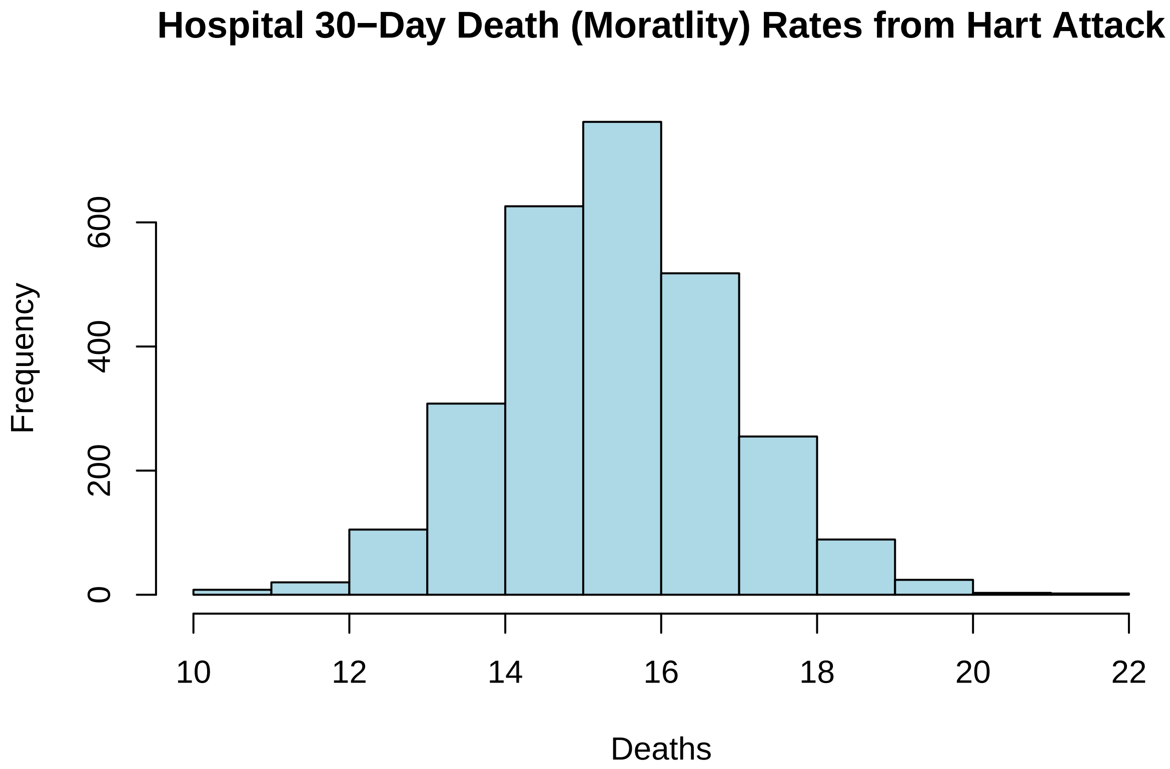
Part 1: Plot the 30-day mortality rates for heart attack

```
outcome <- read.csv("hosp_compare/outcome-of-care-measures.csv", colClasses = "character", header = TRU
head(outcome, 10)
outcome[, 11] <- as.numeric(outcome[, 11], na.rm = TRUE) # set column 11
```

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

```
# NAs introduced by coercion
```

```
hist(outcome[, 11], xlab = "Deaths", main = "Hospital 30-Day Death (Moratlity) Rates from Hart Attack",
```



Part 2: Finding the best hospital in a state

```
# create the function
best <- function(state, outcome) {
  ## Read outcome data
  data <- read.csv("hosp_compare/outcome-of-care-measures.csv", colClasses = "character", header = TRUE
  data2 <- as.data.frame(cbind(data[, 2],   # hospital
                               data[, 7],   # state
                               data[, 11],  # heart attack
                               data[, 17],  # heart failure
                               data[, 23]), # pneumonia
                      stringsAsFactors = FALSE)
  colnames(data2) <- c("hospital", "state", "heart attack", "heart failure", "pneumonia")
  ## Check that state and outcome are valid
  if(!state %in% data2[, "state"]) {
    stop('invalid state')} else if(!outcome %in% c("heart attack", "heart failure", "pneumonia")){
    stop('invalid outcome')
    } else {
    si <- which(data2[, "state"] == state)
    ts <- data2[si, ]    # extracting data for the called state
    oi <- as.numeric(ts[, eval(outcome)])
    min_val <- min(oi, na.rm = TRUE)
    result  <- ts[, "hospital"][which(oi == min_val)]
    output  <- result[order(result)]
  }
  return(output)
}
best("TX", "heart attack")
```

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

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

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

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

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

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

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

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

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

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

Part 3: Ranking hospitals by outcome in a state

```
# create a rankhospital function
rankhospital <- function(state, outcome, rank = "best"){
  ## Read outcome data
  data <- read.csv("hosp_compare/outcome-of-care-measures.csv", colClasses = "character")
  data2   <- as.data.frame(cbind(data[, 2],   # hospital
                              data[, 7],  # state
                              data[, 11],  # heart attack
                              data[, 17],  # heart failure
                              data[, 23]), # pneumonia
                        stringsAsFactors = FALSE)
  colnames(data2) <- c("hospital", "state", "heart attack", "heart failure", "pneumonia")

  ## Check that state and outcome are valid
  if (!state %in% data2[, "state"]) {
    stop('invalid state')
  } else if (!outcome %in% c("heart attack", "heart failure", "pneumonia")){
    stop('invalid outcome')
  } else if (is.numeric(rank)) {
    si <- which(data2[, "state"] == state)
```

```r
    ts <- data2[si, ]                     # extracting dataframe for the called state
    ts[, eval(outcome)] <- as.numeric(ts[, eval(outcome)])
    ts <- ts[order(ts[, eval(outcome)], ts[, "hospital"]), ]
    output <- ts[, "hospital"][rank]
  } else if (!is.numeric(rank)){
    if (rank == "best") {
      output <- best(state, outcome)
    } else if (rank == "worst") {
      si <- which(data2[, "state"] == state)
      ts <- data2[si, ]
      ts[, eval(outcome)] <- as.numeric(ts[, eval(outcome)])
      ts <- ts[order(ts[, eval(outcome)], ts[, "hospital"], decreasing = TRUE), ]
      output <- ts[, "hospital"][1]
    } else {
      stop('invalid rank')
    }
  }
  return(output)
}
# try out
rankhospital("TX", "heart failure", 4)
```

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

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

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

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

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

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

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

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

Part 4: Ranking hospitals in all states

```r
rankall <- function(outcome, num = "best"){
  ## Read outcome data
  data <- read.csv("hosp_compare/outcome-of-care-measures.csv", colClasses = "character")
  data2   <- as.data.frame(cbind(data[, 2],     # hospital
                              data[, 7],  # state
                              data[, 11],  # heart attack
                              data[, 17],  # heart failure
                              data[, 23]), # pneumonia
                        stringsAsFactors = FALSE)
  colnames(data2) <- c("hospital", "state", "heart attack", "heart failure", "pneumonia")
  data2[, eval(outcome)] <- as.numeric(data2[, eval(outcome)])

  ## Check that state and outcome are valid

  if (!outcome %in% c("heart attack", "heart failure", "pneumonia")){
    stop('invalid outcome')
  } else if (is.numeric(num)) {
    by_state <- with(data2, split(data2, state))
    ordered  <- list()
    for (i in seq_along(by_state)){
      by_state[[i]] <- by_state[[i]][order(by_state[[i]][, eval(outcome)],
                                           by_state[[i]][, "hospital"]), ]
      ordered[[i]]  <- c(by_state[[i]][num, "hospital"], by_state[[i]][, "state"][1])
    }
    result <- do.call(rbind, ordered)
    output <- as.data.frame(result, row.names = result[, 2], stringsAsFactors = FALSE)
    names(output) <- c("hospital", "state")
  } else if (!is.numeric(num)) {
    if (num == "best") {
      by_state <- with(data2, split(data2, state))
      ordered  <- list()
      for (i in seq_along(by_state)){
        by_state[[i]] <- by_state[[i]][order(by_state[[i]][, eval(outcome)],
                                             by_state[[i]][, "hospital"]), ]
        ordered[[i]]  <- c(by_state[[i]][1, c("hospital", "state")])
      }
      result <- do.call(rbind, ordered)
      output <- as.data.frame(result, stringsAsFactors = FALSE)
      rownames(output) <- output[, 2]
    } else if (num == "worst") {
      by_state <- with(data2, split(data2, state))
      ordered  <- list()
      for (i in seq_along(by_state)){
        by_state[[i]] <- by_state[[i]][order(by_state[[i]][, eval(outcome)],
                                             by_state[[i]][, "hospital"],
                                             decreasing = TRUE), ]
        ordered[[i]]  <- c(by_state[[i]][1, c("hospital", "state")])
      }
      result <- do.call(rbind, ordered)
      output <- as.data.frame(result, stringsAsFactors = FALSE)
      rownames(output) <- output[, 2]
    } else {
      stop('invalid num')
    }
  }
  return(output)
}
# try out
head(rankall("heart attack", 20), 10)
```

```
## Warning in rankall("heart attack", 20): NAs introduced by coercion
```

```
##                                hospital state
## AK                                 <NA>    AK
## AL      D W MCMILLAN MEMORIAL HOSPITAL    AL
## AR   ARKANSAS METHODIST MEDICAL CENTER    AR
## AZ JOHN C LINCOLN DEER VALLEY HOSPITAL    AZ
## CA               SHERMAN OAKS HOSPITAL    CA
## CO            SKY RIDGE MEDICAL CENTER    CO
## CT             MIDSTATE MEDICAL CENTER    CT
## DC                                 <NA>    DC
## DE                                 <NA>    DE
## FL      SOUTH FLORIDA BAPTIST HOSPITAL    FL
```

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

```
## Warning in rankall("pneumonia", "worst"): NAs introduced by coercion
```

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

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

```
## Warning in rankall("heart failure"): NAs introduced by coercion
```

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