R Programming Project 3

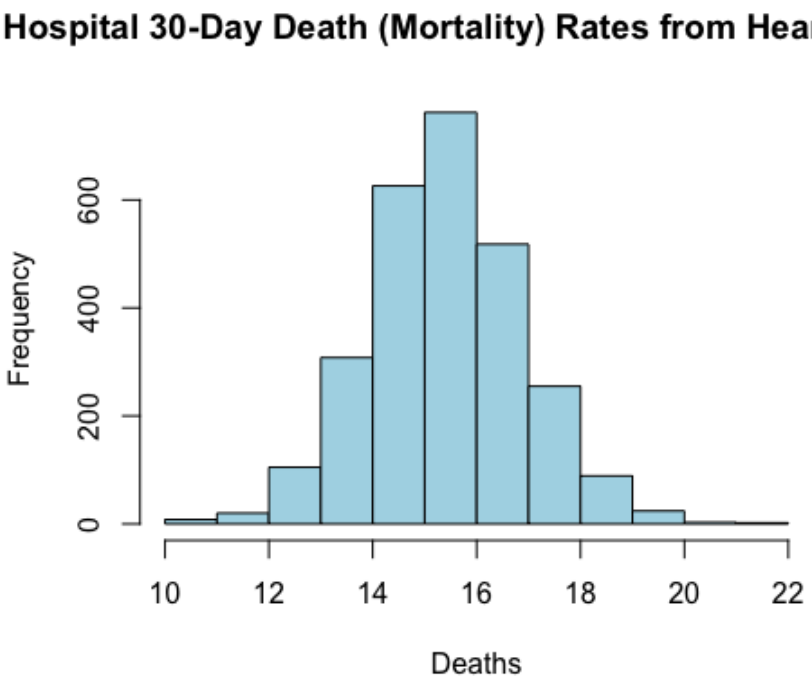
github repo for rest of specialization: Data Science Coursera

The zip file containing the data can be downloaded here: Assignment 3 Data

Part 1 Plot the 30-day mortality rates for heart attack (outcome.R)

```
# install.packages("data.table")
library("data.table")

# Reading in data
outcome <- data.table::fread('outcome-of-care-measures.csv')
outcome[, (11) := lapply(.SD, as.numeric), .SDcols = (11)]
outcome[, lapply(.SD
                 , hist
                 , xlab= "Deaths"
                 , main = "Hospital 30-Day Death (Mortality) Rates from Heart Attack"
                 , col="lightblue")
        , .SDcols = (11)]
```



Part 2 Finding the best hospital in a state (best.R)

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

  # Read outcome data
```

```
  out_dt <- data.table::fread('outcome-of-care-measures.csv')

  outcome <- tolower(outcome)

  # Column name is same as variable so changing it
  chosen_state <- state

  # Check that state and outcome are valid
  if (!chosen_state %in% unique(out_dt[["State"]])) {
    stop('invalid state')
  }

  if (!outcome %in% c("heart attack", "heart failure", "pneumonia")) {
    stop('invalid outcome')
  }

  # Renaming Columns to be less verbose and lowercase
  setnames(out_dt
           , tolower(sapply(colnames(out_dt), gsub, pattern = "^Hospital 30-Day Death \\(Mor
  )

  #Filter by state
  out_dt <- out_dt[state == chosen_state]

  # Columns indices to keep
  col_indices <- grep(paste0("hospital name|state|^",outcome), colnames(out_dt))

  # Filtering out unnessecary data
  out_dt <- out_dt[, .SD ,.SDcols = col_indices]

  # Find out what class each column is
  # sapply(out_dt,class)
  out_dt[, outcome] <- out_dt[,  as.numeric(get(outcome))]


  # Removing Missing Values for numerical datatype (outcome column)
  out_dt <- out_dt[complete.cases(out_dt),]

  # Order Column to Top
  out_dt <- out_dt[order(get(outcome), `hospital name`)]

  return(out_dt[, "hospital name"][1])

}
```

Part 3 Ranking hospitals by outcome in a state (rankhospital.R)

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

  # Read outcome data
  out_dt <- data.table::fread('outcome-of-care-measures.csv')
```

```
  outcome <- tolower(outcome)

  # Column name is same as variable so changing it
  chosen_state <- state

  # Check that state and outcome are valid
  if (!chosen_state %in% unique(out_dt[["State"]])) {
    stop('invalid state')
  }

  if (!outcome %in% c("heart attack", "heart failure", "pneumonia")) {
    stop('invalid outcome')
  }

  # Renaming Columns to be less verbose and lowercase
  setnames(out_dt
           , tolower(sapply(colnames(out_dt), gsub, pattern = "^Hospital 30-Day Death \\(Mor
  )

  #Filter by state
  out_dt <- out_dt[state == chosen_state]

  # Columns indices to keep
  col_indices <- grep(paste0("hospital name|state|^",outcome), colnames(out_dt))

  # Filtering out unnessecary data
  out_dt <- out_dt[, .SD ,.SDcols = col_indices]

  # Find out what class each column is
  # sapply(out_dt,class)
  out_dt[, outcome] <- out_dt[,  as.numeric(get(outcome))]


  # Removing Missing Values for numerical datatype (outcome column)
  out_dt <- out_dt[complete.cases(out_dt),]

  # Order Column to Top
  out_dt <- out_dt[order(get(outcome), `hospital name`)]

  out_dt <- out_dt[,  .(`hospital name` = `hospital name`, state = state, rate = get(outcome

  if (num == "best"){
    return(out_dt[1,`hospital name`])
  }

  if (num == "worst"){
    return(out_dt[.N,`hospital name`])
  }

  return(out_dt[num,`hospital name`])

}
```

Part 4 Ranking hospitals in all states (rankall.R)

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

  # Read outcome data
  out_dt <- data.table::fread('outcome-of-care-measures.csv')

  outcome <- tolower(outcome)

  if (!outcome %in% c("heart attack", "heart failure", "pneumonia")) {
    stop('invalid outcome')
  }

  # Renaming Columns to be less verbose and lowercase
  setnames(out_dt
           , tolower(sapply(colnames(out_dt), gsub, pattern = "^Hospital 30-Day Death \\(Mor
  )

  # Columns indices to keep
  col_indices <- grep(paste0("hospital name|state|^",outcome), colnames(out_dt))

  # Filtering out unnessecary data
  out_dt <- out_dt[, .SD ,.SDcols = col_indices]

  # Find out what class each column is
  # sapply(out_dt,class)

  # Change outcome column class
  out_dt[, outcome] <- out_dt[,  as.numeric(get(outcome))]

  if (num == "best"){
    return(out_dt[order(state, get(outcome), `hospital name`)
    , .(hospital = head(`hospital name`, 1))
    , by = state])
  }

  if (num == "worst"){
    return(out_dt[order(get(outcome), `hospital name`)
    , .(hospital = tail(`hospital name`, 1))
    , by = state])
  }

  return(out_dt[order(state, get(outcome), `hospital name`)
                , head(.SD,num)
                , by = state, .SDcols = c("hospital name") ])

}
```