Assignment 3

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-certified hospitals in the U.S. This dataset essentially covers all major U.S. hospitals. This dataset is used for a variety of purposes, including determining whether hospitals should be fined for not providing high quality care to patients (see http://goo.gl/jAXFX for some background on this particular topic).

Task 1

Plot the 30-day mortality rates for heart attack

Taking a look at the data:

```
outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
head(outcome[1:3,1:5])
```

```
##   Provider.Number                     Hospital.Name                   Address.1
## 1          010001 SOUTHEAST ALABAMA MEDICAL CENTER     1108 ROSS CLARK CIRCLE
## 2          010005    MARSHALL MEDICAL CENTER SOUTH 2505 U S HIGHWAY 431 NORTH
## 3          010006   ELIZA COFFEE MEMORIAL HOSPITAL         205 MARENGO STREET
##   Address.2 Address.3
## 1
## 2
## 3
```

```
str(outcome[,1:10],vec.len=1)
```

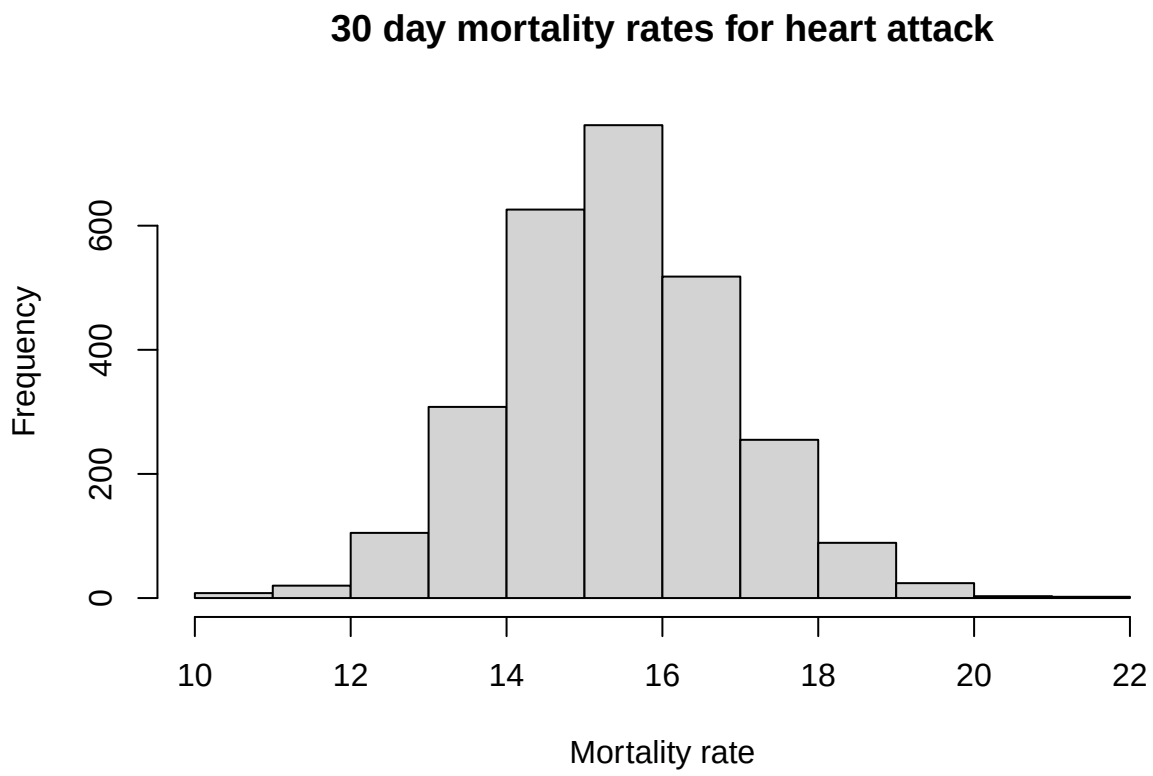
```
## 'data.frame':    4706 obs. of  10 variables:
##  $ Provider.Number: chr  "010001" ...
##  $ Hospital.Name  : chr  "SOUTHEAST ALABAMA MEDICAL CENTER" ...
##  $ Address.1      : chr  "1108 ROSS CLARK CIRCLE" ...
##  $ Address.2      : chr  "" ...
##  $ Address.3      : chr  "" ...
##  $ City           : chr  "DOTHAN" ...
##  $ State          : chr  "AL" ...
##  $ ZIP.Code       : chr  "36301" ...
##  $ County.Name    : chr  "HOUSTON" ...
##  $ Phone.Number   : chr  "3347938701" ...
```

Creating a histogram:

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

```
## Warning:                      NA
```

```r
hist(outcome[, 11],main = "30 day mortality rates for heart attack",xlab="Mortality rate")
```



Task 3 **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).

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".

```r
## best function returns the first hospital (by alphabetical order) with the lowest mortality rate in a
best<- function(state,outcome_name){
  # ''state'' -- the 2-character abbreviated name of a state
  # ''outcome'' -- one of the following ("heart attack", "heart failure", "pneumonia")

  outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
  # First, we define the list of possible states and outcomes. For outcomes, it's also necessary to def

  states_list <- unique(outcome[,7])
```

```
  outcomes_list <- c("heart attack", "heart failure", "pneumonia")
  outcomes_columns <- c("Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack","Hospital.30.Day.Dea
  outcome_col <- NULL

  # Validating the entered values
  if(!state %in% states_list){
    stop("invalid state")
  }
   if(!outcome_name %in% outcomes_list){
    stop("invalid outcome")
   } else {
     # Finding a corresponding column for chosen outcome
    for(j in 1:length(outcomes_list)){
      if(outcome_name==outcomes_list[j]) {outcome_col<-outcomes_columns[j]}
    }
  }

  state_outcome <- subset(outcome,outcome$State==state) # selecting only the hospitals located in the c
  state_outcome[, outcome_col] <- suppressWarnings(as.numeric(state_outcome[, outcome_col])) # changing

  # To deal with ties, we will preliminary sort the data frame by alphabetical order and only then find
  outcome_sorted <- state_outcome[order(state_outcome$Hospital.Name),]
  # Initializing the best score and best name variables
  best_num <- outcome_sorted[[outcome_col]][[1]]
  best_name <- outcome_sorted$Hospital.Name[[1]]
  # Searching for better score in sorted data frame
  for(i in 1:nrow(outcome_sorted)){
    if(!is.na(outcome_sorted[[outcome_col]][[i]]) & (outcome_sorted[[outcome_col]][[i]] < best_num) ){
      best_num<-outcome_sorted[[outcome_col]][[i]]
      best_name<-outcome_sorted$Hospital.Name[[i]]
    }
  }
  best_name # returning the result
}
```

Let's do some testing:

```
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"
```

Let's test on invalid entries:

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

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

```
best("NY", "hert attack")
```

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

The function alone can be found in best.R

Task 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.
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_name, num = "best") {

  # Reading data and checking the entries (similiar to ''best'' function)
  outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
  states_list <- unique(outcome[,7])
  outcomes_list <- c("heart attack", "heart failure", "pneumonia")
  outcomes_columns <- c("Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack","Hospital.30.Day.Dea
  outcome_col <- NULL
  if(!state %in% states_list){
    stop("invalid state")
  }
   if(!outcome_name %in% outcomes_list){
    stop("invalid outcome")
   } else {
    for(j in 1:length(outcomes_list)){
      if(outcome_name==outcomes_list[j]) {outcome_col<-outcomes_columns[j]}
    }
   }
```

```
  state_outcome <- subset(outcome,outcome$State==state,select=c("State","Hospital.Name",outcome_col)) #
  state_outcome[, outcome_col] <- suppressWarnings(as.numeric(state_outcome[, outcome_col])) # changing
  # Sort values and drop NAs
  outcome_sorted <- state_outcome[order(state_outcome[[outcome_col]],state_outcome[["Hospital.Name"]],na

  # Converting character input to numbers
  if(! class(num)=='numeric'){
    if(num == "best") {
      num <- as.numeric(1)
    } else if (num == "worst"){
       num <- as.numeric(nrow(outcome_sorted))
    } else{
      stop("invalid number")
    }
  }

  # In case, if num > the number of rows, return NA
  if(num>nrow(outcome_sorted)){
    return(NA)
  }

  outcome_sorted[num,"Hospital.Name"]
}
```

Testing the code:

```
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
```

All good!

Task 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.

As the assignment stated that we should be using previously written functions, I decided to switch to a different approach and use tapply function on previously sorted dataframe

```
rankall <- function(outcome_name, num = "best") {
  # Reading data and checking the entries (similar to ''best'' function)
  outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")
  outcomes_list <- c("heart attack", "heart failure", "pneumonia")
  outcomes_columns <- c("Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack","Hospital.30.Day.Dea
  outcome_col <- NULL
  if(!outcome_name %in% outcomes_list){
    stop("invalid outcome")
  } else {
    for(j in 1:length(outcomes_list)){
      if(outcome_name==outcomes_list[j]) {outcome_col<-outcomes_columns[j]}
    }
  }
  # drop extra columns
  outcome<-outcome[c("State","Hospital.Name",outcome_col)]
  # sorting data frame by state and by outcome
  outcome[, outcome_col] <- suppressWarnings(as.numeric(outcome[, outcome_col])) # changing values to n
  # Sort values and drop NAs
  outcome_sorted <- outcome[order(outcome[["State"]],outcome[[outcome_col]],outcome[["Hospital.Name"]],n

  if(num=="worst"){
    result<-outcome_sorted[tapply(1:nrow(outcome_sorted),outcome_sorted[["State"]],function(x) tail(x,n=
  } else if (num=="best"){
    result<-outcome_sorted[tapply(1:nrow(outcome_sorted),outcome_sorted[["State"]],function(x) x[1]),]
  } else{
    result<-outcome_sorted[tapply(1:nrow(outcome_sorted),outcome_sorted[["State"]],function(x) x[num]),]
  }


  #result<-outcome_sorted[tapply(1:nrow(outcome_sorted),outcome_sorted[["State"]],function(x) tail(x,n=
  # The problem is that results contains NAs where num is higher than the number of hospitals in the st

  states_list <- unique(outcome_sorted[["State"]])
  idx <- which(is.na(result$State), arr.ind = TRUE)
  result[idx,1]<-states_list[idx]
  # dropping indexes for prettier output
  rownames(result) <- NULL
  result
}
```

Let's do some testing:

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

```
##    State                         Hospital.Name
## 1     AK                                 <NA>
```

```
## 2     AL      D W MCMILLAN MEMORIAL HOSPITAL
## 3     AR   ARKANSAS METHODIST MEDICAL CENTER
## 4     AZ JOHN C LINCOLN DEER VALLEY HOSPITAL
## 5     CA               SHERMAN OAKS HOSPITAL
## 6     CO            SKY RIDGE MEDICAL CENTER
## 7     CT             MIDSTATE MEDICAL CENTER
## 8     DC                                <NA>
## 9     DE                                <NA>
## 10    FL      SOUTH FLORIDA BAPTIST HOSPITAL
##    Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack
## 1                                                        NA
## 2                                                      15.7
## 3                                                      17.1
## 4                                                      14.9
## 5                                                      13.3
## 6                                                      15.0
## 7                                                      15.6
## 8                                                        NA
## 9                                                        NA
## 10                                                     13.5
```

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

```
##    State                              Hospital.Name
## 52    WI MAYO CLINIC HEALTH SYSTEM - NORTHLAND, INC
## 53    WV                      PLATEAU MEDICAL CENTER
## 54    WY           NORTH BIG HORN HOSPITAL DISTRICT
##    Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia
## 52                                                   17.4
## 53                                                   16.2
## 54                                                   17.3
```

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

```
##    State                                                Hospital.Name
## 45    TN                          WELLMONT HAWKINS COUNTY MEMORIAL HOSPITAL
## 46    TX                                    FORT DUNCAN MEDICAL CENTER
## 47    UT VA SALT LAKE CITY HEALTHCARE - GEORGE E. WAHLEN VA MEDICAL CENTER
## 48    VA                                     SENTARA POTOMAC HOSPITAL
## 49    VI                            GOV JUAN F LUIS HOSPITAL & MEDICAL CTR
## 50    VT                                        SPRINGFIELD HOSPITAL
## 51    WA                                   HARBORVIEW MEDICAL CENTER
## 52    WI                                AURORA ST LUKES MEDICAL CENTER
## 53    WV                                     FAIRMONT GENERAL HOSPITAL
## 54    WY                                   CHEYENNE VA MEDICAL CENTER
##    Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure
## 45                                                       9.4
## 46                                                       8.1
## 47                                                      10.7
## 48                                                       8.4
## 49                                                      13.9
## 50                                                      10.9
## 51                                                       8.9
```

```
## 52                                                  9.3
## 53                                                  9.5
## 54                                                 10.3
```

All done!