

Ranking of Hospitals based on Mortality Rates

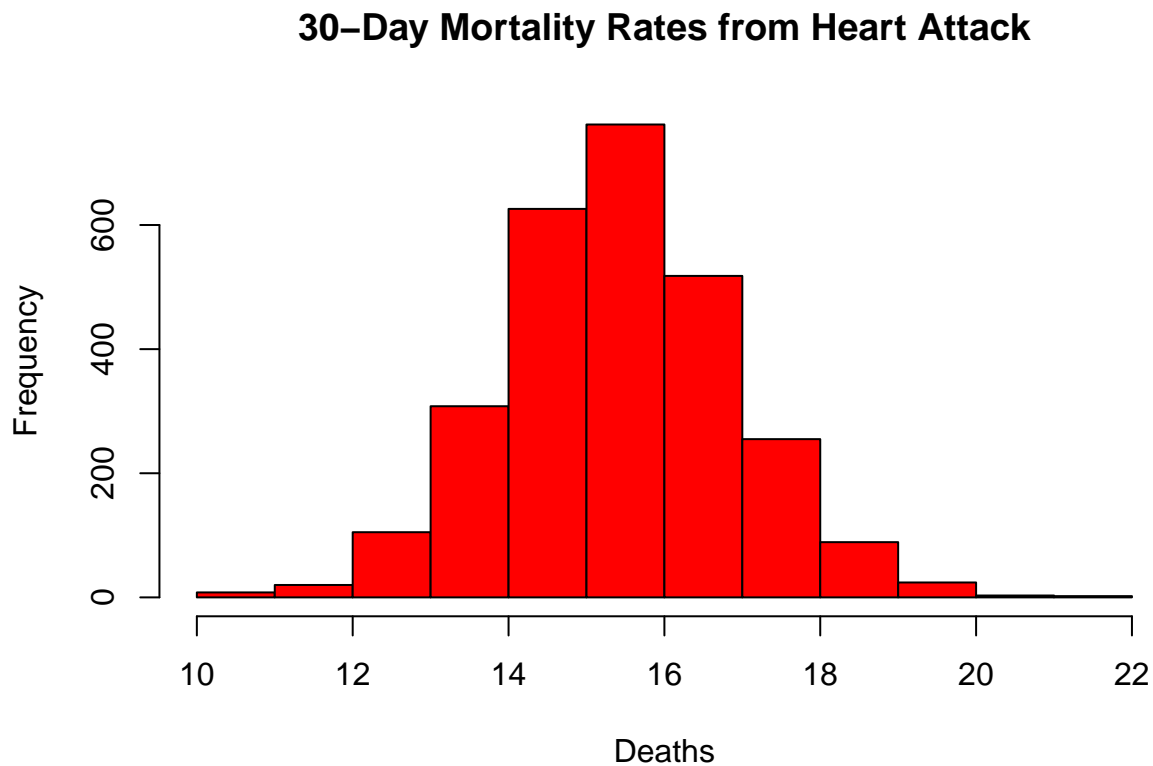
PART 1: Plot the 30-day mortality rates for heart attack

```
outcome <- read.csv("outcome-of-care-measures.csv", colClasses = "character")

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

## Warning: NAs introduced by coercion
## You may get a warning about NAs being introduced; that is okay

hist(outcome[, 11], main="30-Day Mortality Rates from Heart Attack",
      xlab="Deaths",col="red")
```



PART 2: Finding the best hospital in a state

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.

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

```

## Read outcome data
data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")

#create new dataframe with required columns
df <- as.data.frame(cbind(data[,2],
                           data[,7],
                           data[,11],
                           data[,17],
                           data[,23]),
                    stringsAsFactors = FALSE)

colnames(df) <- c("hospital", "state", "heart attack", "heart failure", "pneumonia")

chosen_state <- state

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

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

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

else {
  x <- which(df[, "state"] == chosen_state)
  y <- df[x,]

  vals <- as.numeric(y[, eval(outcome)]) #convert into numeric type

  min_val <- min(vals, na.rm = TRUE)

  hos <- y[which(vals == min_val), "hospital"]

}

hos

}

best("TX", "heart attack")

```

```

## Warning in best("TX", "heart attack"): NAs introduced by coercion
## [1] "CYPRESS FAIRBANKS MEDICAL CENTER"

```

PART 3: Ranking hospitals by outcome in a state

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 and the disease specified by the outcome argument.

```

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

  ## Read outcome data
  data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")

  #create new dataframe with required columns
  df <- as.data.frame(cbind(data[,2],
                             data[,7],
                             data[,11],
                             data[,17],
                             data[,23]),
                     stringsAsFactors = FALSE)

  colnames(df) <- c("hospital", "state", "heart attack", "heart failure", "pneumonia")

  chosen_state <- state

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

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

  ## Return hospital name in that state with the given rank
  ## 30-day death rate
  else if(is.numeric(num)) {
    x <- which(df[, "state"] == chosen_state)
    y <- df[x,] #new df filtered by state

    y[,eval(outcome)] <- as.numeric(y[,eval(outcome)]) #convert into numeric type
    z <- y[order(y[,eval(outcome)], y[, "hospital"]),]
    result <- z[, "hospital"][num]
  }
  else if (!is.numeric(num)){
    if(num == "best"){
      result <- best(state, outcome)
    }
    else if(num == "worst"){
      x <- which(df[, "state"] == chosen_state)
      y <- df[x,] #new df filtered by state

      y[,eval(outcome)] <- as.numeric(y[,eval(outcome)]) #convert into numeric type
      z <- y[order(y[,eval(outcome)], y[, "hospital"], decreasing = TRUE),]
      result <- z[, "hospital"][1]
    }
    else{
      stop("invalid rank")
    }
  }
  result
}

```

```
rankhospital("TX", "heart failure", 4)
```

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

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

PART 4: Ranking hospitals in all states

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.

```
rankall <- function(outcome, num = "best") {  
  
  ## Read outcome data  
  data <- read.csv("outcome-of-care-measures.csv", colClasses = "character")  
  
  #Get distinct states from the data  
  states <- levels(factor(data[,7]))  
  
  #create new dataframe with required columns  
  df <- as.data.frame(cbind(data[,2], #hospital name  
                             data[,7], #state  
                             data[,11], #heart attack  
                             data[,17], #heart failure  
                             data[,23]), #pneumonia  
                      stringsAsFactors = FALSE)  
  
  colnames(df) <- c("hospital", "state", "heart attack", "heart failure", "pneumonia")  
  
  #Possible Outcomes  
  outcomes <- c("heart attack", "heart failure", "pneumonia")  
  
  #Check validity of input of state, outcome and num  
  if(!outcome %in% outcomes){  
    stop("invalid outcome")  
  }  
  else if(is.numeric(num)){  
    by_state <- with(df, split(df, 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(df, split(df, 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(df, split(df, 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)
}

```

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

##		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