Coursera R Week 4 programming assignment 3

Mike Fang

March 13, 2016

I'd say this is a tougher project than I thought it was, giving my experience in statistical analysis with R.Nonetheless, I managed to finish the assignment without any issues, and my functions ran very smoothly.

The function could be improved further with some fine tuning, but I'd say it is good enough for passing the course.

The first function finds best hospital in state

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

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

# example output:
best("SC", "heart attack")
```

```
## [1] "MUSC MEDICAL CENTER"
```

```
# it will give a warning message for converting char strings to numerics, I hide it here
```

The second funtion ranks hospitals by outcome in a state

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

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

# example output:
rankhospital("NC", "heart attack", "worst")
```

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

The third function ranks hospitals in all states.

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

```r
# example output:
r <- rankall("heart attack", 4)
as.character(subset(r, state == "HI")$hospital)
```

```
## [1] "CASTLE MEDICAL CENTER"
```

```r
head(rankall("heart attack", "worst"))
```

```
##                             hospital state
## AK   MAT-SU REGIONAL MEDICAL CENTER    AK
## AL   HELEN KELLER MEMORIAL HOSPITAL    AL
## AR    MEDICAL CENTER SOUTH ARKANSAS    AR
## AZ      VERDE VALLEY MEDICAL CENTER    AZ
## CA METHODIST HOSPITAL OF SACRAMENTO    CA
## CO    NORTH SUBURBAN MEDICAL CENTER    CO
```