

R Programming - Assignment #3

Myron Keith Gibert Jr

January 21, 2020

Contents

Correspondence	1
Introduction	1
Set Parameters	1
Debug	1
Data	2
Unzipping the data	2
Plot the 30-day mortality rates for heart attack	2
Finding the best hospital in a state	3
Ranking hospitals by outcome in a state	5
Ranking hospitals in all states	6
Quiz	7
Cleanup	9

Correspondence

Please address any questions to Myron Keith Gibert Jr at mkgibertjr@msn.com. Code for this project is stored in a [GitHub repository](#).

Introduction

For this assignment, I created two functions. The first function will create a special list vector that can store an inverted matrix based on an input matrix. The second function will create an inverted matrix using the special vector and store that matrix within that vector's environment. In addition to acquiring more practice in writing functions, this assignment involved an understanding of lexical scoping. Because of this, I will take extra steps to explain what each function does, and then perform some tests at the end to confirm that the functions work as intended.

The full instructions for this assignment can be found [here](#).

Set Parameters

```
#set output directory? Default: outputdir <- "assignment3outputs"  
outputdir <- "assignment3outputs"  
#Overwrite contents of the output directory? Default: deleteoutputs <- FALSE  
deleteoutputs <- TRUE  
#Delete specdata/ directory after completing the analysis? Default: deletespec <- TRUE  
deletehos <- TRUE
```

Debug

The debug chunk will prevent the script from running if any of the dependent variables for this analysis do not exist. This should prevent the program from erroring out after a long runtime without producing any results due to a missing variable. If modifying the input .csv and .xlsx files, it is important to leave all header information and column names intact, as the program uses this information to extract relevant data. Columns are intuitively labeled to end user convenience.

```
if (dir.exists(outputdir) && deleteoutputs == FALSE ){  
  stop("Your output directory already exists! Please delete/move  
    this folder from your working directory. Alternatively, you  
    can set 'deleteoutputs' to TRUE to auto-delete this folder  
    for every run. You may also choose an alternative output  
    directory.")  
}  
else{  
  unlink(outputdir,recursive = TRUE)  
}  
  
if (!exists("outputdir")){  
  stop("outputdir variable is not defined. Please ensure that all  
    parameters in the r parameters chunk are defined.")  
}  
  
if (!exists("deleteoutputs")){  
  stop("deleteoutputs variable is not defined. Please ensure that all  
    parameters in the r parameters chunk are defined.")  
}  
  
if (!exists("deletehos")){
```

```
stop("deletehos variable is not defined. Please ensure that all
      parameters in the r parameters chunk are defined.")
}

if (!dir.exists(outputdir)){dir.create(outputdir)}
```

Data

Unzipping the data

For this programming assignment I needed to unzip this folder. Once I unzipped the zip file, I did not make any modifications to the files in the directories.

```
if(
  !dir.exists("hospital-data.csv") |
  !dir.exists("outcome-of-care-measures.csv") |
  !dir.exists("Hospital_Revised_Flatfiles.pdf")
){unzip("ASN3_rprog_data_hospital.zip")}
```

Plot the 30-day mortality rates for heart attack

I first needed to read the outcome data into R via the read.csv function and look at the first few rows.

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

head(outcomedata)
```

I then need to make a simple histogram of the 30-day death rates from heart attack. This is column 11 in the outcome dataset.

```
# I will get a warning about NAs being introduced; that is okay
outcomedata[,11] <- as.numeric(outcomedata[,11])

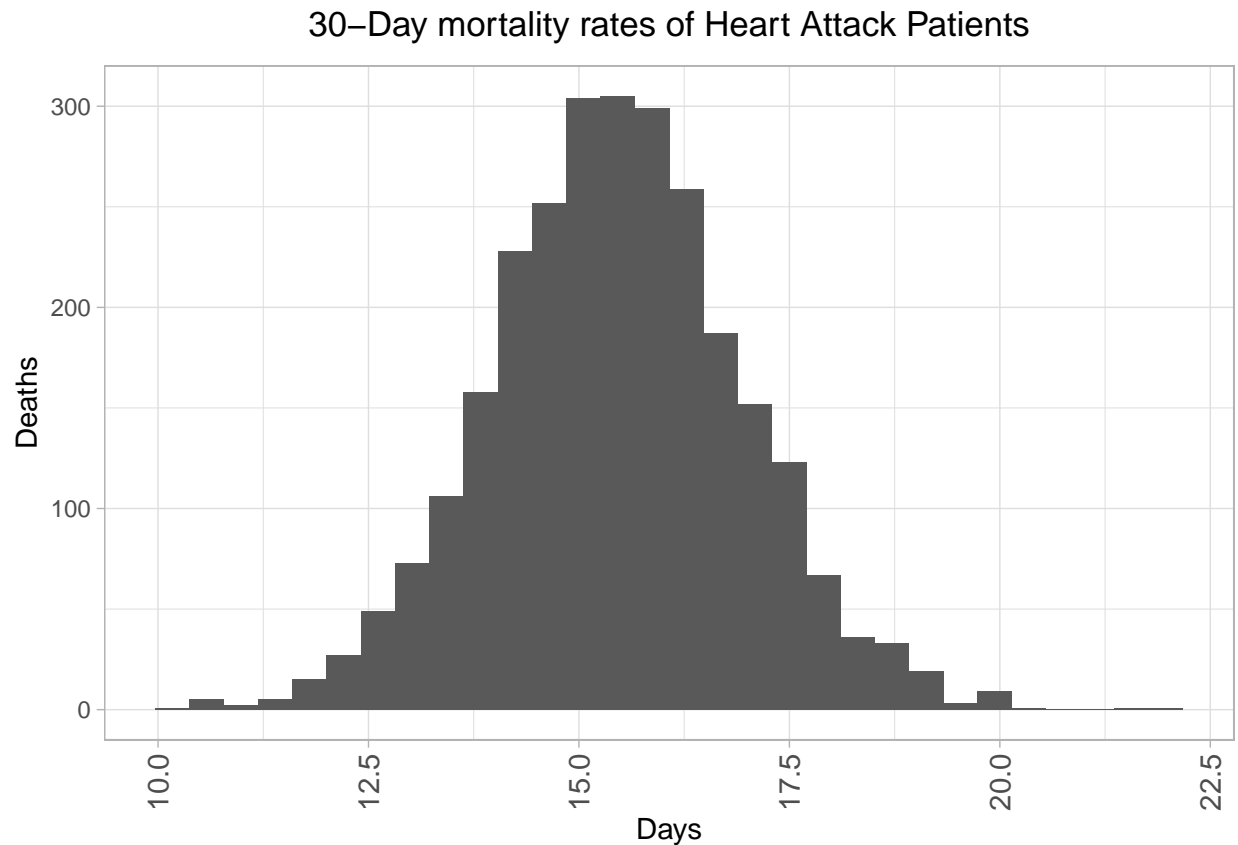
heartattack <- outcomedata[,11]

heartattack <- as.data.frame(heartattack[complete.cases(heartattack)])

colnames(heartattack) <- "thirty_day_HA_mortality"

p <- ggplot(heartattack,aes(thirty_day_HA_mortality)) +
  geom_histogram() +
  xlab("Days") +
  ylab(expression("Deaths")) +
  ggtitle(expression("30-Day mortality rates of Heart Attack Patients")) +
  theme_light() +
  scale_color_colorblind() +
  scale_fill_colorblind() +
  theme(axis.text.x=element_text(size=rel(1.2),angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(size=rel(1.2), face="bold",hjust = 0.5),
        legend.position = "none")

p
```



Because I originally read the data in as character (by specifying `colClasses = "character"`), I need to coerce the column to be numeric. I got a warning about NAs being introduced but that is okay, as there is missing data.

Finding the best hospital in a state

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

  #state <- "TX"
  #outcome <- "heart attack"

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

  if (is.na(match(state,data$State))){
    stop("invalid state")
  }

  if (outcome == "heart attack" |
      outcome == "Heart Attack" |
      outcome == "heart failure" |
      outcome == "Heart Failure" |
      outcome == "pneumonia" |
      outcome == "Pneumonia"){else{
    stop("invalid outcome")
  }
}
```

```

beststate <- data[which(data$State==state),]

if(outcome=="heart attack" | outcome=="Heart Attack"){
  minrate <- min(
    as.numeric(
      beststate$Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack)
    ,na.rm=TRUE)
  besthos <- beststate$Hospital.Name[
    match(
      minrate
      ,as.numeric(
        beststate$Hospital.30.Day.Death..Mortality..Rates.from.Heart.Attack))]
}else{
  if(outcome=="heart failure" | outcome=="Heart Failure"){
    minrate <- min(
      as.numeric(
        beststate$Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure)
      ,na.rm=TRUE)
    besthos <- beststate$Hospital.Name[
      match(
        minrate
        ,as.numeric(
          beststate$Hospital.30.Day.Death..Mortality..Rates.from.Heart.Failure))]
  }else{
    if(outcome=="pneumonia" | outcome=="Pneumonia"){
      minrate <- min(
        as.numeric(
          beststate$Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia)
        ,na.rm=TRUE)
      besthos <- beststate$Hospital.Name[
        match(
          minrate
          ,as.numeric(
            beststate$Hospital.30.Day.Death..Mortality..Rates.from.Pneumonia))]
    }
  }

  besthos

}

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"

```

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

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

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

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

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

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

Ranking hospitals by outcome in a state

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

  #state <- "MD"
  #outcome <- "heart attack"
  #num <- "worst"

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

  if (is.na(match(state,data$State))){
    stop("invalid state")
  }

  if (outcome == "heart attack" |
      outcome == "Heart Attack"){i <- 11} else if(
      outcome == "heart failure" |
      outcome == "Heart Failure"){i <- 17} else if(
      outcome == "pneumonia" |
      outcome == "Pneumonia"){i <- 23}else{
    stop("invalid outcome")
  }

  beststate <- data[which(data$State==state),]
  beststate[,i] <- as.numeric(beststate[,i])

  beststate <- beststate[which(!is.na(beststate[,i])),]

  beststate <- beststate[order(beststate[,i], beststate$Hospital.Name), ]

  if(num=="best" | num=="Best"){
    num2 <- 1
  }

  if(num=="worst" | num=="Worst"){
    num2 <- nrow(beststate)
  }else if(is.numeric(num)) {num2 <- num}
  else if(num<1 || num > nrow(beststate)) {
    return(NA)
  } else {
    stop('invalid num')
  }
}
```

```

    }

return.names <- beststate[num2, ]$Hospital.Name

return.names[1]

}

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

```

Ranking hospitals in all states

```

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

  #outcome <- "pneumonia"
  #num <- "worst"

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

  if (outcome == "heart attack" |
      outcome == "Heart Attack"){i <- 11} else if(
      outcome == "heart failure" |
      outcome == "Heart Failure"){i <- 17} else if(
      outcome == "pneumonia" |
      outcome == "Pneumonia"){i <- 23}else{
    stop("invalid outcome")
  }

  data[,i] <- as.numeric(data[,i])

  state.i <- 1

  states <- unique(as.character(data$State))
  ranking <- data.frame(hospital=character(0),state=character(0))

  for (state.i in 1:length(states)){
    #state <- "WY"
    state <- states[state.i]

    data.state <- data[which(data$State==state),]

    data.state[,i] <- as.numeric(data.state[,i])

    data.state <- data.state[which(!is.na(data.state[,i])),]
  }
}

```

```

data.state <- data.state[order(data.state[,i], data.state$Hospital.Name), ]

if(num=="best" | num=="Best"){
  num2 <- 1
}

if(num=="worst" | num=="Worst"){
  num2 <- nrow(data.state)
}else if(is.numeric(num)) {num2 <- num}
  else if(num<1 || num > nrow(data.state)) {
    return(NA)
  } else {
    stop('invalid num')
  }

return.names <- data.state[num2, ]$Hospital.Name

hospital <- return.names[1]

rbinder <- cbind(state,hospital)

ranking <- rbind(ranking,rbinder)
}
colnames(ranking) <- c("state","hospital")
ranking[order(ranking$state),]
}

test1 <- rankall("heart attack", 20)
head(test1,10)

```

```

##      state                                hospital
## 2      AK                                <NA>
## 1      AL      D W MCMILLAN MEMORIAL HOSPITAL
## 4      AR      ARKANSAS METHODIST MEDICAL CENTER
## 3      AZ      JOHN C LINCOLN DEER VALLEY HOSPITAL
## 5      CA      SHERMAN OAKS HOSPITAL
## 6      CO      SKY RIDGE MEDICAL CENTER
## 7      CT      MIDSTATE MEDICAL CENTER
## 9      DC                                <NA>
## 8      DE                                <NA>
## 10     FL      SOUTH FLORIDA BAPTIST HOSPITAL

```

```

test2 <- rankall("pneumonia","worst")
tail(test2,3)

```

```

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

```

Quiz

After completing the programming assignment, I had to complete a quiz using the provided code with each question. I first had to run the provided code, and then select my output from the multiple choice options.

This quiz was used to verify whether my three functions can effectively answer the data science questions from this data set. My final score was a 100%.

1. What result is returned by the following code?

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

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

2. What result is returned by the following code?

```
best("NY", "pneumonia")
```

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

3. What result is returned by the following code?

```
best("AK", "pneumonia")
```

```
## [1] "YUKON KUSKOKWIM DELTA REG HOSPITAL"
```

4. What result is returned by the following code?

```
rankhospital("NC", "heart attack", "worst")
```

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

5. What result is returned by the following code?

```
rankhospital("WA", "heart attack", 7)
```

```
## [1] "YAKIMA VALLEY MEMORIAL HOSPITAL"
```

6. What result is returned by the following code?

```
rankhospital("TX", "pneumonia", 10)
```

```
## [1] "SETON SMITHVILLE REGIONAL HOSPITAL"
```

7. What result is returned by the following code?

```
rankhospital("NY", "heart attack", 7)
```

```
## [1] "BELLEVUE HOSPITAL CENTER"
```

8. What result is returned by the following code?

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

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

9. What result is returned by the following code?

```
r <- rankall("pneumonia", "worst")
as.character(subset(r, state == "NJ")$hospital)
```

```
## [1] "BERGEN REGIONAL MEDICAL CENTER"
```

10. What result is returned by the following code?

```
r <- rankall("heart failure", 10)
as.character(subset(r, state == "NV")$hospital)
```

```
## [1] "RENOWN SOUTH MEADOWS MEDICAL CENTER"
```

Cleanup

This final command removes the unzipped “specdata” directory if the `deletespec` variable is set to `TRUE`. This reduces the overall storage burden of this project by removing the files that we no longer need access to. The zipped file remains in the working directory, so “specdata” will be recreated anyways using the command in line 93 (`unzip`) if it is deleted here.

```
if(deletehos== TRUE){  
  file.remove("hospital-data.csv")  
  file.remove("outcome-of-care-measures.csv")  
  file.remove("Hospital_Revised_Flatfiles.pdf")  
}
```

```
## [1] TRUE
```