Yexuan Gu **UP 494** Lab Exercise 4 11/8/16

Creating Graphics and Dashboards with R

Tools

- Csv files of daily jail log data from http://dart.ncsa.uiuc.edu/stuffed/bpnj/
- R and its packages: shiny, shinydashboard, highcharter, dplyr, tidyr, rdrop2, lubridate, RCurl

Procedures

```
install.packages(
                                                                                     install.packages("shinydashboard")
                                                                                     install.packages("highcharter")
install.packages("dplyr")
                                                                                    install.packages("tidyr")
install.packages("rdrop2")
install.packages("lubridate")
install.packages("RCurl")
                                                                                   library(shiny)
library(shinydashboard)
library(highcharter)
library(dplyr)|
library(tidyr)
library(rdrop2)
library(lubridate)
library(RCurl)
1) Install and library packages
```

- 2) Set work directory to my work folder

```
setwd("C:\\Users\\ygu23\\Desktop\\Academic\\UP494\\LabExercise4")
                                   data <- read.csv('djlsummary.all_LE4.csv')</pre>
                                   data$Date <- as.Date(data$custDate, format="%m/%d/%Y")</pre>
                                   y.limit <- max(data$popTot)</pre>
                                   output.table <- data
                                  Chg.Tot.Pop <- diff(data$popTot)
                                   daye <- lubridate::wday(data$custDate, label=T, abbr = F)</pre>
Read and organize data
         k <- list()
```

```
30 • for (i in 1:((length(daye)) - 1)) {
31     k[i] <- (pasteO(daye[i], " to ", daye[i+1], ""))
    Chg.Tot.Pop.Labels <- as.character(k)
change.data <- cbind.data.frame(Chg.Tot.Pop.Labels, Chg.Tot.Pop)
    colnames(change.data ) <- c("Period", "Change in Total Population")</pre>
    lval <- dim(data)[1]</pre>
    lval <- lval * -:
    data.sorted <- data[order(-1:lval),]</pre>
```

- 4) Set up user interface
 - a) Dashboard page includes dashboard header, sidebar, sidebar menu. The first step is to set up

```
<- dashboardPage
                             dashboardHeader(title = "Champaign County Jail Population Summary"),
                             dashboardSidebar(
                             sidebarMenu(
                                 menuItem("Dashboard", tabName = "Dashboard"),
menuItem("Raw Data", tabName = "Rawdata")
these elements
```

b) Still in the "dashboardPage" command, we need to set up the dashboard body based on the csv

file that we collected. There are two items within the dashboard body: "dashboard" and "raw data". Within the first tab item, dashboard, there is the majority of our data visualization. We use fluidRow to visualize the data into boxes. The data includes total jail population today, total inmates sentenced, total inmates end of sentence, total jail population distributed by location, satellite and downtown, change in jail population, total bookings yesterday, total inmates transferred, total inmates released yesterday, total inmates with appointments and total inmates with appointments / bond.

```
dashboardBody(
          tabItems(
               tabItem("Dashboard",
                  fluidRow(
                     valueBoxOutput("total"),
                     valueBoxOutput("sentenced"),
valueBoxOutput("unsentenced"),
valueBoxOutput("end-of-sentence")
17
18
                 ),
fluidRow(
                     box(
                        width = 8, status = "info", solidHeader = TRUE,
                        title = "Distribution by Location",
23
24
                        highchartOutput("plot1")
25
26
27
28
29
30
                     box(
                        width = 4, status = "info",
title = "Change in Jail Population",
                        tableOutput("table1")
                  ),
fluidRow(
                     valueBoxOutput("bookings"),
                    valueBoxOutput("transfers"),
valueBoxOutput("releases"),
valueBoxOutput("appts"),
valueBoxOutput("apptom")
33
34
```

c) In Raw Data, we also enable users to download all data as csv.

```
tabItem("Rawdata",
numericInput("maxrows", "Rows to show", 25, min=1, max=dim(data)[1]),
verbatimTextOutput("rawtable"),
downloadButton("downloadCsv", "Download All Data as CSV")
```

- 5) After setting up the user interface, we need to script the server.
 - a) Using Highchart to visualize the total jail population distributed in downtown and satellite.

b) Render the chart we just made with specific attributes

```
15  output$total <- renderValueBox({
16    valueBox(
17    value = data.sorted$popTot[1],
18    subtitle = "Total Jail Population Today",
19    icon = icon("area-chart"),
20    coplor = "yellow"
21    )
22  })</pre>
```

c) Set the box for other information, such as total jail population, total inmates sentenced, etc. Give different colors to boxes with different information.

```
output$sentenced <- renderValueBox({</pre>
       output$total <- renderValueBox({</pre>
         valueBox(
                                                                    valueBox(
           value = data.sorted$popTot[1],
                                                                      value = data.sorted$custSent[1],
                                                                      subtitle =
           subtitle = "Total Jail Population Today",
                                                                                   "Total Inmates Sentenced",
                                                                      icon = icon("balance-scale"),
color = "light-blue"
           icon = icon("area-chart"),
           color = "yellow"
                                                            output$unsentenced <- renderValueBox({</pre>
      output$unsentenced <- renderValueBox({</pre>
                                                              valueBox(
                                                                value = data.sorted$custEndSent[1],
34
           value = data.sorted$custUnsent[1],
                                                                icon = icon("clock-o"),
           icon = icon("clock-o"),
                                                                color = "blue
           color = "aqua'
      output$bookings <- renderValueBox({</pre>
                                                              output$transfers <- renderValueBox({</pre>
        valueBox(
                                                                valueBox(
          value = data.sorted$bookings[1],
                                                                  value = data.sorted$transfers[1],
54
55
          subtitle = "Total Bookings Yesterday",
icon = icon("book"),
                                                                  subtitle = "Total Inmates Transferred",
                                                                  icon = icon("exchange"),
           color = "yellow"
                                                                  color = "light-blue
                                                           output$appts <- renderValueBox({
      output$releases <- renderValueBox({</pre>
                                                              valueBox(
        valueBox(
                                                                value = data.sorted$appts[1],
          value = data.sorted$releases[1],
                                                                 Total Inmates With Appointments",
            Total Inmates Released Yesterday"
          icon = icon("sign-out"),
                                                                icon = icon("sign-out"),
                                                                color = "teal'
          color = "aqua"
       output$apBond <- renderValueBox({</pre>
         valueBox(
            value = data.sorted$apBond[1],
            "Total Inmates With Appointments/Bond", icon = icon("sign-out"),
            color = "olive"
```

d) Create the table for change in jail population

```
outputStable1 <- renderTable({
    lval <- dim(change.data)[1]
    lval <- lval * -1
    change.data.display <- change.data.display, 11)
    }, digits = 1, include.rownames=FALSE)

output$downloadCsv <- downloadHandler(
    filename = "djlsummary.al]_LE4.csv",
    content = function(file) {
    write.csv(data, file)
    },
    contentType = "text/csv"

    load
    load
```

e) Output the raw data into a table in the raw data tab.

```
110 output$rawtable <- renderPrint({
111 orig <- options(width = 1000)
112 print((tail(data, input$maxrows)),row.names=FALSE)
113 options(orig)
114 })
115 }
```

Results and Analysis

Link on shinyapp.io: http://127.0.0.1:5759

Figure 1 and Figure 2 show the final visualization production of the shiny dashboard based on the daily jail log summary data. Based on the Total Population Distribution by Location chart, it can be easily concluded that there are approximately 1/3 jail population are booked from downtown area. The chart is clear for users to get the proportion of specific data and compare them with others. The eight boxes shown explicitly the real-time intraday population data, working perfectly for daily jail data. The table of Change in Jail Population indicates the change of data in different period, which gives clear comparisons among different periods for users. In addition, raw data page makes the data more accessible for users to have a look and download directly.

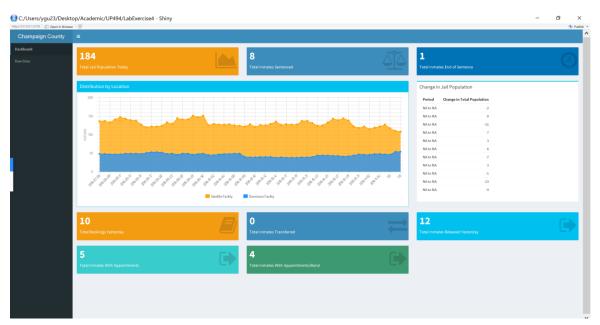


Figure 1. Dashboard page for daily jail population in Champaign County

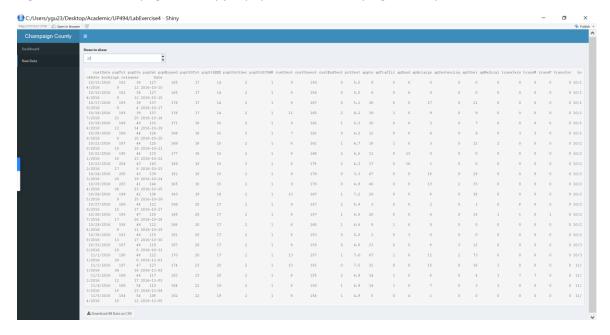


Figure 2. Raw data page for daily jail population in Champaign County

Discussion

In my opinion, the shiny dashboard package is tremendously useful and applicable for urban planning. Firstly, government data are always in pdf or excel files. Even though government encourages and makes efforts to make the data transparent to residents, the complicated forms of data still impede the users' reading and understanding. The shiny dashboard increases the readability of the data from diverse professional institutions not only for residents, but also for planners, designers, engineers, etc. Secondly, shiny dashboard not only simply visualize the data in its original formats, but it also helps to organize the data into specific form for different purposes of use. For example, the figure made by Highchart visualizes the data into a comparison between different items. Thirdly, there will be new data coming in the system fast every day. The data can be updated fast by these tools to meet changes and uncertainties for residents' uses and urban planning.