Shiny App (Main)

Nivishan

12/02/2021

```
#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(sf)
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```r
library(tmap)

source("code/loadData.R")
source("code/loadMap.R")


crimeLA <- loadData()
shpLA <- loadMap()

crimeLA  <- crimeLA %>%
    mutate(dateReported = date(mdy_hms(dateReported)),
           dateOccurred = date(mdy_hms(dateOccurred)),
           timeOccurred =  format(strptime(timeOccurred,format = "%H%M"),'%H:%M'),
           hour = hour(hm(timeOccurred)), ## Extract hour from time
           dayOfWeek =  wday(dateOccurred,label = T) )

LAPDDivisions <- crimeLA %>% distinct(areaName) %>% pull(areaName)
LAPDDivisions
```

```
##  [1] "Southwest"   "Central"     "Van Nuys"    "N Hollywood" "Mission"
##  [6] "Devonshire"  "Hollywood"   "Harbor"      "Olympic"     "Northeast"
## [11] "Southeast"   "West Valley" "Pacific"     "Wilshire"    "West LA"
## [16] "Foothill"    "Newton"      "77th Street" "Rampart"     "Hollenbeck"
## [21] "Topanga"
```

```r
topCrimeLables <- c("VEHICLE - STOLEN" = "Motor vehicle theft",
                    "BATTERY - SIMPLE ASSAULT" = "Battery",
                    "VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)" = "Vandalism - Felony ",
                    "BURGLARY FROM VEHICLE" = "Burglary from vehicle",
                    "BURGLARY" = "Burglary",
                    "ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT"  = "Aggravated Assault",
                    "INTIMATE PARTNER - SIMPLE ASSAULT" = "Simple Assault",
                    "THEFT PLAIN - PETTY ($950 & UNDER)" = "Petty Theft",
                    "THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)" = "Theft from motor vehicle",
                    "VANDALISM - MISDEAMEANOR ($399 OR UNDER)" = "Vandalism - Misdemeanor"
)


# Define UI for application that draws a histogram
ui <- fluidPage(

    # Application title
    titlePanel("Los Angeles Crime Dashboad"),

    # Sidebar with a slider input for number of bins
    sidebarLayout(
        sidebarPanel(
            selectInput(inputId = "LAPDDivSelect",
                        label = "Select LAPD Divisions",
                        choices = LAPDDivisions,
                        multiple = T,
                        selected = "Central"),
            dateRangeInput(inputId = "timePeriod",
                           label = "Select time period",
                           start = "2020-01-01",
                           end = "2020-12-01",
                           min = "2020-01-01",
                           max = "2020-12-12"),
        width = 2),

        # Show a plot of the generated distribution
        mainPanel(
            fluidRow(
                column(5,
                       tmapOutput(outputId = "crimeMap",height = 700)),
                column(7,
                       fluidRow(
                           column(6, plotOutput(outputId = "victAgeHist")),
                           column(6, tableOutput(outputId = "crimeTable") )
                       ),
                       fluidRow(plotOutput(outputId = "crimeTimeSeries"))
                ),


            ),width = 10) # end of mainPanel
    )
) # End of fluidPage


    # Define server logic required to draw a histogram
    server <- function(input, output) {

        crimeData <- reactive({
            crimeLA %>% filter(areaName %in% input$LAPDDivSelect,
                               dateOccurred >= input$timePeriod[1],
                               dateOccurred <=   input$timePeriod[2] )
        })

        output$victAgeHist <- renderPlot({
            crimeData()  %>% filter(victAge !=0) %>%
                select(victAge) %>%
                ggplot() +
                geom_histogram(mapping = aes(x=victAge),binwidth = 10) +
                xlab("Age of victim") +
                theme_classic()
        })

        output$crimeMap <- renderTmap({

            crimePerRptDist <-  crimeData()%>%
                group_by(areaName,rptDistNum) %>%
                summarise(count= n()) %>%
                mutate(rptDistNum = as.integer(rptDistNum))


            left_join(shpLA,crimePerRptDist, by = c("REPDIST" = "rptDistNum" )) %>%
                filter(areaName %in% input$LAPDDivSelect ) %>%
                tm_shape() +
                tm_polygons("count",style="jenks")  +
                tm_layout(legend.position = c("left", "bottom"),
                          legend.title.size = 1,
                          legend.text.size = 0.8)

        }
        )

        output$crimeTable <- renderTable({
            topCrimes <- crimeData() %>%
                group_by(crimeCodeDesc) %>%
                summarise(count = n()) %>%
                arrange(desc(count)) %>%
                head(n = 10) %>%
                pull(crimeCodeDesc)

            topCrimesCount <-  crimeData() %>%
                filter(crimeCodeDesc %in% topCrimes) %>%
                group_by(crimeCodeDesc) %>%
                summarise(Count = n()) %>%
                arrange(desc(Count)) %>%
                rename(Crime = crimeCodeDesc)

            otherCrimeCount <- nrow(crimeData()) - sum(topCrimesCount$count)

            topCrimesCount %>%
                add_row(tibble_row(Crime = "Other", Count = otherCrimeCount))
        }, bordered = T)

        output$crimeTimeSeries  <- renderPlot({

             topCrimes <- crimeData() %>%
                group_by(crimeCodeDesc) %>%
                summarise(count = n()) %>%
                arrange(desc(count)) %>%
                head(n = 5) %>%
                pull(crimeCodeDesc)

            topCrimesCount <-  crimeData() %>%
                filter(crimeCodeDesc %in% topCrimes) %>%
                group_by(dateOccurred,crimeCodeDesc) %>%
                summarise(count = n())

            ggplot(topCrimesCount) +
                geom_line(mapping = aes(x = dateOccurred ,y=count, color = crimeCodeDesc),stat = "identi
                scale_color_discrete(labels=topCrimeLables, name="Crime")  +
                xlab("Date") +
                ylab("Number of incidents") +
                theme_classic()
        })
    }
```

```r
# Run the application
shinyApp(ui = ui, server = server)
```