PreliminaryResults_ForPeerReview

Luke Wilde

4/28/2022

Contents

Goals and Topic

Mapping terrestrial migrations has recently emerged as an essential tool for their conservation (Kauffman et al. 2021). An improved knowledge of the whereabouts of animal populations and key aspects of their habitats is increasing as anthropogenic influences endanger terrestrial communities (Harris et al. 2009). Map making can require large, often interdisciplinary teams to improve the usability and aesthetics of their design (Bolger et al. 2007), posing a problem for many researchers with limited experience. Tools to seamlessly connect datastreams with highly technical data cleaning processes with the rest of mapping teams are essential for productivity (Zhang and Li 2005). Shiny apps have proved to be powerful tools for web-based applications that use large datasets and require seamless integration of computer programming with the user experience (Elliott and Elliott 2020).

The application will include two basic components to fulfill this goal: first, users will have a interactive and customizable map to view the movement data from any number of animals according to migration timing or a specific date range (see below *Dataset Overview* for more details); second, users will be able to export the mapped data to a .csv, .shp, or .kml file types for easy integration with any number of mapping software or field applications. Users will be the collaborative team working with the Red Desert mule deer project, my current PhD project. Without a tidy means of sharing data with collaborators, miscommunication and outdated data have become common (Wang, Cook, and Hyndman 2020). Thus, *my main goal is to develop a web-based Shiny application to allow collaborators of the Red Desert mule deer project to access, explore and download data.*

Begin Implemented Workflow

The below workflow shows the creation of a user interface and server for shiny. I am not sure if this will run as a Markdown since Shiny apps need the `app.R` file to be able to run. Therefore, I am simply placing the ui and server here, and going to use eval=F to show the code but not run it. To run the app, please use this link to access my Github repo. https://github.com/luke-wilde/Wilde_DataViz_BuildingShinyForMapDataSharing

Note: This code does not include the data pre-processing steps that I used to collect and format the data. See app.R for more.

```
#load or install
library(tidyverse)
library(lubridate)
library(sf)
library(shiny)
library(shinyWidgets)
library(shinydashboard)
library(leaflet)
```

Loading the data and briefly formatting

The data are a subset of mule deer monitored in 2019 and 2020. Random scatter was applied to maintain anonymity. The important columns are ***AID*** - *animal ID numbers* `Mgtry` - migratory strategy/distance (Long, Medium, and Short) ***POSIXct*** - *datetime stamp* `Year` - year, four digit *`season` - coded based on ranges: these are spring and fall migration and winter and summer ranges.

```
#data is only from a sample of deer, daily locations from satellite service (Proprietary reasons)

getwd() #make sure this is the app.R file location
```

```
## [1] "C:/Users/14064/Documents/GitHub/Wilde_DataViz_BuildingShinyForMapDataSharing/Wilde_ShinyAppForDa
```

```
#load the data table and shp file
gps.data <- read.csv("./Shiny/muledeer_test_data.csv")
gps.sf <- st_read("./Shiny/muledeer_test_data.shp")
```

```
## Reading layer 'muledeer_test_data' from data source
##   'C:\Users\14064\Documents\GitHub\Wilde_DataViz_BuildingShinyForMapDataSharing\Wilde_ShinyAppForData
##   using driver 'ESRI Shapefile'
## Simple feature collection with 2336 features and 17 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 543353 ymin: 4616996 xmax: 723862 ymax: 4818232
## CRS:           unknown
```

```
#ensure that the date time is correct, cannot be a character when using date sliders
gps.sf$POSIXct <- as.POSIXct(paste(gps.sf$POSIXct, gps.sf$Time, " "), format= "%Y-%m-%d %H:%M:%S", tz="
```

```
#check if over
head(gps.data)
```

```
##   AID  Mgtry TagColor TagNum CllrSrN Frequency     Lat   Long
## 1 426 Medium    White      1 706084A    162.09 4626960 661426
## 2 426 Medium    White      1 706084A    162.09 4635808 656332
## 3 426 Medium    White      1 706084A    162.09 4637314 660456
## 4 426 Medium    White      1 706084A    162.09 4652290 667612
## 5 426 Medium    White      1 706084A    162.09 4657141 664992
## 6 426 Medium    White      1 706084A    162.09 4650963 670649
```

```
##               POSIXct       Date     Time Year            SpringMS
## 1 2019-06-24 17:00:00 2019-06-24 17:00:00 2019 2019-06-24 03:00:00
## 2 2019-06-25 17:00:00 2019-06-25 17:00:00 2019 2019-06-24 03:00:00
## 3 2019-06-26 17:00:00 2019-06-26 17:00:00 2019 2019-06-24 03:00:00
## 4 2019-06-27 17:00:00 2019-06-27 17:00:00 2019 2019-06-24 03:00:00
## 5 2019-06-28 17:00:00 2019-06-28 17:00:00 2019 2019-06-24 03:00:00
## 6 2019-06-29 17:00:00 2019-06-29 17:00:00 2019 2019-06-24 03:00:00
##              SpringME              FallMS              FallME           season
## 1 2019-07-22 07:00:00 2019-10-22 05:00:00 2019-10-26 05:00:00 Spring Migration
## 2 2019-07-22 07:00:00 2019-10-22 05:00:00 2019-10-26 05:00:00 Spring Migration
## 3 2019-07-22 07:00:00 2019-10-22 05:00:00 2019-10-26 05:00:00 Spring Migration
## 4 2019-07-22 07:00:00 2019-10-22 05:00:00 2019-10-26 05:00:00 Spring Migration
## 5 2019-07-22 07:00:00 2019-10-22 05:00:00 2019-10-26 05:00:00 Spring Migration
## 6 2019-07-22 07:00:00 2019-10-22 05:00:00 2019-10-26 05:00:00 Spring Migration
```

```
#needs to be in latlong for leaflet to work
gps.sf <- st_transform(gps.sf, "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
```

Creating the user interface

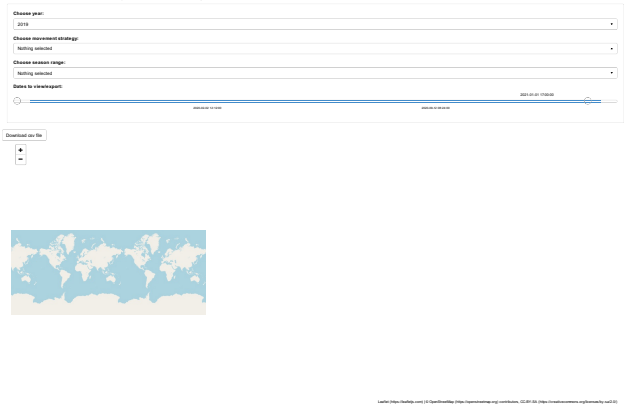
```
shinyApp(
# Define UI for application
ui = fluidPage(
  titlePanel(title="Red Desert Mule Deer Project -- Data sharing tool"),
  sidebarLayout(
    sidebarPanel(
      pickerInput("Year", "Choose year:",choices=unique(gps.sf$Year), multiple=F, options=list('actions
      pickerInput("Mig", "Choose movement strategy:",choices=unique(gps.sf$Mgtry), multiple=T, options=
      pickerInput("Type", "Choose season range:", choices=unique(gps.sf$season), multiple=T, options=lis
                sliderInput("range",
                   "Dates to view/export:",
                   min=min(gps.sf$POSIXct, na.rm = TRUE),
                   max=max(gps.sf$POSIXct, na.rm = TRUE),
                   value = range(gps.sf$POSIXct, na.rm = FALSE),
                   step = 1)
       ),
   # actionButton("download_shp", "Download shp file"),
    actionButton("download_csv", "Download csv file")
  ),
  mainPanel(leafletOutput("map", width="100%", height=800))
),

#define server with reactive functionality
server = function(input, output, session) {
  filter_year <- reactive(
    gps.sf %>% filter(Year %in% input$Year)
  )

  observeEvent(input$Year, {
    updatePickerInput(
      session=session,
      inputId = "Mig",
```

```
      choices = unique(filter_year()$Mgtry)
    )
  })

  filter_mig <- reactive(
    filter_year() %>% filter(Mgtry %in% input$Mig)
  )

  observeEvent(input$Mig, {
    updatePickerInput(
      session=session,
      inputId = "Type",
      choices = unique(filter_mig()$season)
    )
  })

  filter_type <- reactive(
    filter_mig() %>% filter(season %in% input$Type)
  )

  observeEvent(input$Type,{
    updateSliderInput(
      session = session,
      inputId = "range",
      min = min(filter_type()$POSIXct),
      max = max(filter_type()$POSIXct),
      value = range(filter_type()$POSIXct, na.rm = FALSE)
    )
  }
)

  filter_range <- reactive(
    filter_type() %>% filter(POSIXct >= input$range[1] & POSIXct <= input$range[2])
  )

  observe({ pal <- colorFactor(palette = 'Dark2', domain = filter_range()$AID)

  output$map <- renderLeaflet({
    leaflet(gps.sf) %>%
      addTiles() %>%
      addCircleMarkers(data = filter_range(), color="grey", fillColor = ~pal(AID),fillOpacity = 1, popu
  })})

  data <- reactive(
    filter_range()
  )

  output$download_csv <- downloadHandler(
    filename = function(){
      paste(getwd(),"/RDMD_datadownload_",str_replace_all(Sys.Date(),pattern="-",replacement=""),".csv"
    },
    content = function(file){
      write.csv(as.data.frame(filter_range()), file, row.names=F)
```

```
    }
  )

  # output$download_shp <- downloadHandler(
  #   filename=paste("RDMD_datadownload_",str_replace_all(Sys.Date(),pattern="-",replacement=""),sep="",
  #
  #   content = function(file) {
  #     data = filter_range() # I assume this is a reactive object
  #     # create a temp folder for shp files
  #     temp_shp <- getwd()
  #     # write shp files
  #     st_write(data, paste(filename,".shp",sep=""),append=F)
  #     # zip all the shp files
  #     zip_file <- file.path(temp_shp, paste(filename,".zip",sep=""))
  #     shp_files <- list.files(temp_shp,
  #                             filename,
  #                             full.names = TRUE)
  #     # the following zip method works for me in linux but substitute with whatever method working in
  #     zip_command <- paste("zip -j",
  #                          zip_file,
  #                          paste(shp_files, collapse = " "))
  #     system(zip_command)
  #     # copy the zip file to the file argument
  #     file.copy(zip_file, file)
  #     # remove all the files created
  #     file.remove(zip_file, shp_files)
  #   }



},

 options = list(height = 500)
# launch the app locally
# shinyApp(ui, server)
)
```

```
##
## Listening on http://127.0.0.1:4584
```

Red Desert Mule Deer Project -- Data sharing tool



References

Bolger, Douglas T., William D. Newmark, Thomas A. Morrison, and Daniel F. Doak. 2007. "The Need for Integrative Approaches to Understand and Conserve Migratory Ungulates." *Ecology Letters* 0 (0): 070926060247001–??? https://doi.org/10.1111/j.1461-0248.2007.01109.x.

Elliott, Matthew S., and Lisa M. Elliott, eds. 2020. *Developing R Shiny Web Applications for Extension Education*.

Harris, Grant, Simon Thirgood, J. Grant C. Hopcraft, Joris P. G. M. Cromsigt, and Joel Berger. 2009. "Global Decline in Aggregated Migrations of Large Terrestrial Mammals." *Endangered Species Research* 7 (1): 55–76. https://doi.org/10.3354/esr00173.

Kauffman, Matthew J., Francesca Cagnacci, Simon Chamaillé-Jammes, Mark Hebblewhite, J. Grant C. Hopcraft, Jerod A. Merkle, Thomas Mueller, et al. 2021. "Mapping Out a Future for Ungulate Migrations." *Science* 372 (6542): 566–69. https://doi.org/10.1126/science.abf0998.

Wang, Earo, Dianne Cook, and Rob J. Hyndman. 2020. "A New Tidy Data Structure to Support Exploration and Modeling of Temporal Data." *Journal of Computational and Graphical Statistics* 29 (3): 466–78. https://doi.org/10.1080/10618600.2019.1695624.

Zhang, Chuanrong, and Weidong Li. 2005. "The Roles of Web Feature and Web Map Services in Real-Time Geospatial Data Sharing for Time-Critical Applications." *Cartography and Geographic Information Science* 32 (4): 269–83. https://doi.org/10.1559/152304005775194728.

Github

https://github.com/luke-wilde/Wilde_DataViz_BuildingShinyForMapDataSharing