Homework 7

Name: Put your name here

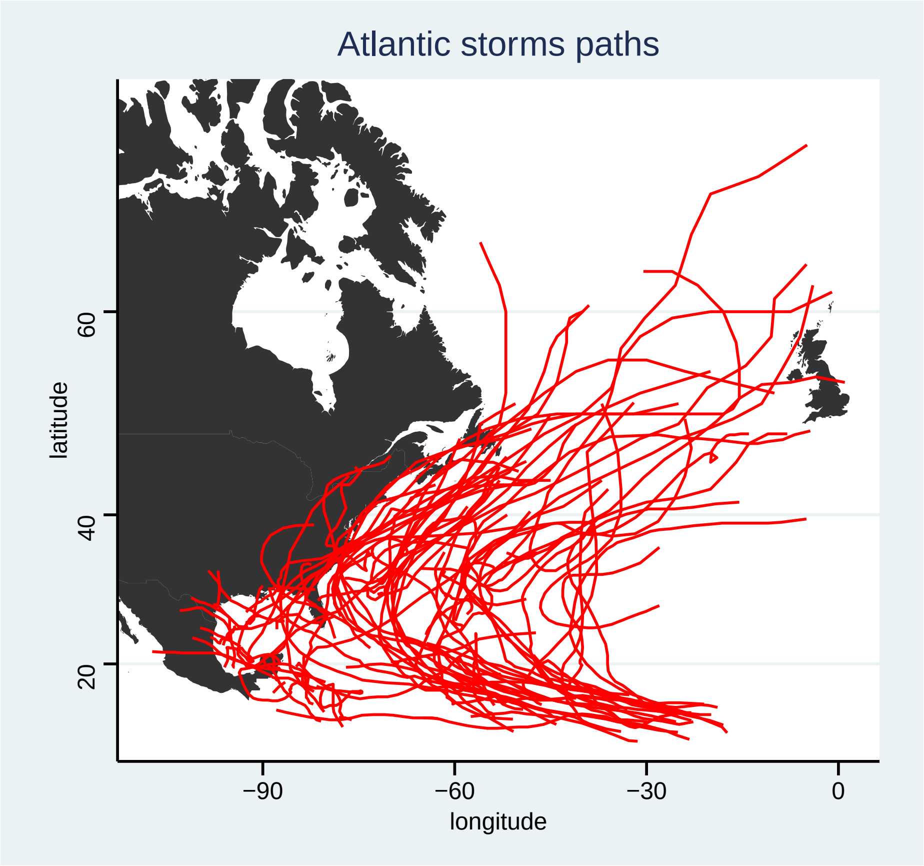
I worked with:

Click the “Knit” button in RStudio to knit this file to a pdf.

Probelm 1: Storm paths

```r
data(storms, package = "nasaweather")
ctry <- map_data("world",
                 region = c(
                   "usa",
                   "mexico",
                   "canada",
                   "uk"
                 ))
base_map <- ggplot(ctry) +
  geom_polygon(aes(x = long,  y = lat, group = group)) +
  labs(
    x = "longitude",
    y = "latitude",
    title = "Atlantic storms paths"
    )

base_map +
  geom_path(data = storms, aes(x = long, y = lat, group = name), color = "red") +
  coord_map(xlim  = c(min(storms$long), max(storms$long)),
            ylim  = c(min(storms$lat), max(storms$lat)))
```



a.

answer:

b.

answer:

c.

answer:

Problem 2: Shiny App for Visualizing Baby Name Trends

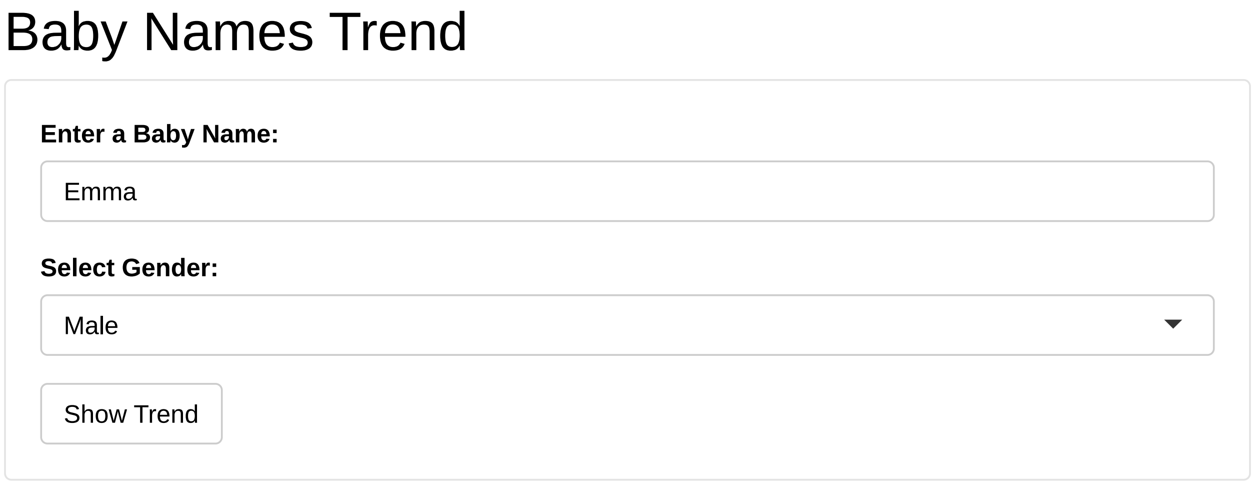
In this assignment, as seen in class activity 19, you will extend a basic Shiny app to visualize trends in baby names. The foundation of the app includes a user interface for input and a server setup to filter data reactively. Your task is to expand this app to allow for the visualization of multiple baby names on the same plot, updating reactively with the press of an action button.

```r
ui <- fluidPage(
  titlePanel("Baby Names Trend"),
  sidebarLayout(
    sidebarPanel(
      textInput("name", "Enter a Baby Name:", "Emma"),
      selectInput("gender", "Select Gender:", choices = c("Male" = "M", "Female" = "F")),
      actionButton("goButton", "Show Trend")
    ),
    mainPanel(plotOutput("nameTrend"))
  )
)
```

```
server <- function(input, output) {
  nameData <- eventReactive(input$goButton, {
    req(input$name) # Ensure the name input is not empty
    babynames %>%
      filter(name == isolate(input$name), sex == isolate(input$gender))
  })


  output$nameTrend <- renderPlot({
    req(nameData())
    ggplot(nameData(), aes(x = year, y = n)) +
      geom_line() +
      labs(title = paste("Trend for name", isolate(input$name)),
           x = "Year", y = "Number of Babies") +
      theme_minimal()
  })
}

shinyApp(ui, server)
```


Baby Names Trend

Enter a Baby Name: Emma

Select Gender: Male

a.

answer:

b.

answer:

```
# app url here
```

Problem 3: Spam using k-nn

This example looks at a data set of about 4600 emails that are classified as spam or not spam, along with over 50 variables measuring different characteristic of the email. Details about these variables are found on the Spambase example on the machine learning data archive. The dataset linked to below is a slightly cleaned up version of this data. The only extra column in the data is `rgroup` which is a randomly assigned grouping variable (groups 0 through 99) which we will eliminate from the data.

Read the data in using the commands below to create a response `class` variable that contains the factor levels `spam` and `nonspam` with `spam` the first level.

```
# tsv = tab separated values!
spam <- read_delim("http://math.carleton.edu/kstclair/data/spamD.txt",
        delim="\t")

# some clean up
spam <- spam %>%
  mutate(class = fct_recode(
    spam,
    spam = "spam" ,
    nonspam = "non-spam"), # rename levels because caret doesn't like "non-spam"
    class = fct_relevel(class, "spam") # make "spam" the first level (our "positive")
    ) %>%
  select(-rgroup, -spam) # don't need random group variable and  spam variable
levels(spam$class)  # verify "spam" is level 1
## [1] "spam"    "nonspam"
```

a.

Answer: Your answer here

```
# Your code here
```

b.

```
set.seed(757302859)  # set a seed
```

Answer: Your answer here

```
# Your code here
```

c.

Make a recipe for fitting k nearest-neighbor algorithm to the training data by inputting the formula and the preprocessing steps.

Answer: Your answer here

```
# Your code here
```

d.

Answer: Your answer here

```
# Your code here
```

e.

Answer: Your answer here

```
# Your code here
```

f.

Answer: Your answer here

```
# Your code here
```

g.

Answer: Your answer here

```
# Your code here
```

h.

Use the `tidymodels` package to do 10-fold cross validation as follows:

- use the 80% training data split from part b.
- tune your knn spam classifier based on accuracy
- consider neighborhood sizes ranging from size 1 to 31

Use the `results` to get the training set cross-validated estimates of the accuracy, precision, sensitivity and specificity of your final ("best") classifier.

And use the following seed before running your `train` command:

```
set.seed(30498492)
```

Answer: Your answer here

```
# Your code here
```

i.

Answer: Your answer here

```
# Your code here
```

Problem 4: Incoming student characteristic

We will look at a "classic" college data set of a random sample of colleges and universities. To simplify our look at this data, we will filter to only look at MN, MA, and CA schools

```
colleges <- read_csv("http://math.carleton.edu/kstclair/data/Colleges.csv")
names(colleges)
##  [1] "State"       "College"     "SATM"        "SATV"        "AppsReceive"
##  [6] "AppsAccept"  "HStop10"     "HStop25"     "FullTime"    "Tuition"
## [11] "RoomBoard"   "Books"       "Ratio"       "Donate"      "Expend"
## [16] "GradRate"    "Type"        "AvgSalary"   "NumFaculty"
colleges2 <- colleges %>%
  filter(State %in% c("MN","MA","CA"))
colleges2 %>% count(State)
## # A tibble: 3 x 2
##   State     n
##   <chr> <int>
## 1 CA       21
## 2 MA       19
## 3 MN       11
```

We will also just focus on student body characteristics (incoming class averages) for SAT and the HS variables (which are are the proportion of the incoming class that is in the top 10% or 25% of their HS class). Here we select just these characteristics and college name and state.

```
colleges2 <- colleges2 %>% select(1,2,3,4,7,8)
colleges2
## # A tibble: 51 x 6
##    State College                           SATM  SATV HStop10 HStop25
##    <chr> <chr>                            <dbl> <dbl>   <dbl>   <dbl>
##  1 CA    California Institute of Technolo   750   660      98     100
##  2 CA    California Lutheran University     495   436      23      52
##  3 CA    California Polytechnic-San Luis    547   455      47      73
##  4 CA    Chapman University                 501   456      23      48
##  5 CA    Claremont McKenna College          670   600      71      93
##  6 CA    Harvey Mudd College                740   630      95     100
##  7 CA    Pitzer College                     590   560      37      73
##  8 CA    Pomona College                     700   640      80      98
##  9 CA    Scripps College                    590   560      60      83
## 10 CA    Occidental College                 570   510      52      81
## # i 41 more rows
```

Let's cluster schools by their incoming class characteristics.

(a)

Answer: Your answer here

```
# Your code here
```

(b)

Answer: Your answer here

```
# Your code here
```

(c)

Answer: Your answer here

```
# Your code here
```

(d)

Answer: Your answer here

```
# Your code here
```

(e)

Answer: Your answer here

```
# Your code here
```

(f)

Answer: Your answer here

```
# Your code here
```

(g)

```
library(GGally)   # install if needed
colleges2 %>%
  ggpairs(aes(color = cluster_km3),
          columns=c("SATM", "SATV", "HStop10", "HStop25"))
```

Answer: Your answer here

```
# Your code here
```