

# Interactive Postsecondary Institution Map

## Introduction and goal

The aim of this project is to represent in an interactive environment the dataset “school” in “CollegeScorecard”. Precisely, only data with latitude and longitude available are represented. This interactive interface is based on the location of the postsecondary school institution on the map.

## Libraries and dataset

Firstly, libraries and dataset are going to be loaded. If necessary, some packages need to be installed.

```
# Install necessary packages
# install.packages(c("collegeScorecard", "leaflet", "leaflet.extras", "dplyr", "shiny"))

# Load required libraries
library(collegeScorecard) # Dataset containing information about U.S. colleges
library(leaflet)          # For interactive maps
library(leaflet.extras)  # Additional features for Leaflet maps
library(dplyr)            # Data manipulation
library(shiny)            # Web application framework for R

# Load College Scorecard dataset
data("school")
```

- collegeScorecard: provides the dataset of postsecondary institutions.
- leaflet & leaflet.extras: enable dynamic map creation and interactive elements.
- dplyr: helps to filter and manipulate data efficiently.
- shiny: framework for creating web-based interactive applications.

## Data preparation

In this section, the variable “state” is associated with its real name to facilitate the interaction with the Shiny interface. Also, variables with no data for longitude and latitude are discarded because of the goal of the project. Finally, states are divided in states if they belong to the USA and territories if they do not.

```
# Create a mapping of state abbreviations to full state names
state_names <- data.frame(
  state = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
            "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
            "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
            "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
            "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY",
```

```

      "AS", "GU", "MP", "PR", "VI", "PW", "FM", "MH"),
full_name = c("Alabama", "Alaska", "Arizona", "Arkansas", "California",
              "Colorado", "Connecticut", "Delaware", "Florida", "Georgia",
              "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa", "Kansas",
              "Kentucky", "Louisiana", "Maine", "Maryland", "Massachusetts",
              "Michigan", "Minnesota", "Mississippi", "Missouri", "Montana",
              "Nebraska", "Nevada", "New Hampshire", "New Jersey",
              "New Mexico", "New York", "North Carolina", "North Dakota",
              "Ohio", "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island",
              "South Carolina", "South Dakota", "Tennessee", "Texas", "Utah",
              "Vermont", "Virginia", "Washington", "West Virginia", "Wisconsin",
              "Wyoming", "American Samoa", "Guam", "Northern Mariana Islands",
              "Puerto Rico", "U.S. Virgin Islands", "Palau",
              "Federated States of Micronesia", "Marshall Islands")
)

# Merge university data with full state names
college_data <- school %>%
  filter(!is.na(latitude) & !is.na(longitude) & !is.na(state)) %>%
  left_join(state_names, by = "state")

# Separate U.S. states and territories into different lists
us_states <- state_names %>%
  filter(!(full_name %in% c("Puerto Rico", "American Samoa", "Guam", "Northern Mariana Islands", "U.S. Virgin Islands", "Palau", "Federated States of Micronesia", "Marshall Islands")))
pull(full_name)

territories <- state_names %>%
  filter(full_name %in% c("Puerto Rico", "American Samoa", "Guam", "Northern Mariana Islands", "U.S. Virgin Islands", "Palau", "Federated States of Micronesia", "Marshall Islands"))
pull(full_name)

```

- state\_names: maps state abbreviations to full names to enrich the dataset.
- filter(!is.na(...)): removes rows with missing location information (latitude, longitude, state).
- left\_join: merges the dataset with the state names for better visualization.

## Define user interface (UI)

In this part, the user interface is created in order to have a map to visualize the data and a dropdown menu to choose between states or territories.

```

ui <- fluidPage(
  titlePanel("Interactive Postsecondary Institution Map"),

  # Dropdown menu positioned at the top
  fluidRow(
    column(12,
      selectInput(
        "state",
        "Select a State or Territory:",
        choices = list(
          "All" = "All",
          "U.S. States" = us_states,
          "Territories" = territories

```

```

    ),
    selected = "All"
  )
)
),
# Map centered below
fluidRow(
  column(12,
    leafletOutput("map", height = "600px") # Adjust the height for better visualization
  )
)
)

```

- fluidPage: provides the layout for the web application.
- selectInput: dropdown menu for users to choose between displaying “All”, specific U.S. states, or territories
- leafletOutput: placeholder for rendering the interactive map.

## Define server logic

In this section, the specifics of the server are programmed. A heatmap with culsters is configured. Also the popup are configured as follows: name of institution, city, website and state or territory, depending on the case. These variable are provided in all the data with longitude and latitude, so there is need to hide these categories in any case. Nevertheless, highest degree offered is shown when available, otherwise, it is shown predominant degree offered.

```

server <- function(input, output, session) {
  output$map <- renderLeaflet({
    if (input$state == "All") {
      # Heatmap for all institutions, when user choose All in the menu
      heatmap_data <- college_data
      leaflet(data = heatmap_data) %>%
        addTiles() %>%
        addProviderTiles(providers$CartoDB.Positron) %>%
        setMaxBounds(
          lng1 = -180, lat1 = -60, # Southwest corner of bounds
          lng2 = 180, lat2 = 85   # Northeast corner of bounds
        ) %>%
        addHeatmap(
          lng = ~longitude,
          lat = ~latitude,
          blur = 20,
          max = 0.05,
          radius = 15
        ) %>%
        addMarkers(
          lng = ~longitude,
          lat = ~latitude,
          popup = ~paste0(
            "<b>Institution:</b> ", name, "<br>",
            ifelse(!is.na(deg_highest),

```

```

        paste0("<b>Highest degree offered:</b> ", as.character(deg_highest), "<br>"),
        paste0("<b>Predominant degree offered:</b> ", as.character(deg_predominant), "<br>")
    ),
    "<b>Website:</b> <a href='", url, "' target='_blank'>", url, "</a><br>",
    "<b>Location:</b> ", city, "<br>",
    ifelse(full_name %in% us_states,
        paste0("<b>State:</b> ", full_name, " (", state, ")<br>"),
        paste0("<b>Territory:</b> ", full_name, " (", state, ")<br>")
    )
  ),
  clusterOptions = markerClusterOptions()
)
} else {
  # Map with individual markers for a selected state or territory
  filtered_data <- college_data %>% filter(full_name == input$state)
  buffer <- 0.1

  if (nrow(filtered_data) > 0) {
    leaflet(data = filtered_data) %>%
      addTiles() %>%
      addProviderTiles(providers$CartoDB.Positron) %>%
      setMaxBounds(
        lng1 = -180, lat1 = -60,
        lng2 = 180, lat2 = 85
      ) %>%
      addMarkers(
        ~longitude,
        ~latitude,
        popup = ~paste0(
          "<b>Institution:</b> ", name, "<br>",
          ifelse(!is.na(deg_highest),
            paste0("<b>Highest degree offered:</b> ", as.character(deg_highest), "<br>"),
            paste0("<b>Predominant degree offered:</b> ", as.character(deg_predominant), "<br>")
          ),
          "<b>Website:</b> <a href='", url, "' target='_blank'>", url, "</a><br>",
          "<b>Location:</b> ", city, "<br>",
          ifelse(full_name %in% us_states,
            paste0("<b>State:</b> ", full_name, " (", state, ")<br>"),
            paste0("<b>Territory:</b> ", full_name, " (", state, ")<br>")
          )
        )
      ) %>%
      #This is added to show the smallest and remote areas in the map
      fitBounds(
        min(filtered_data$longitude, na.rm = TRUE) - buffer,
        min(filtered_data$latitude, na.rm = TRUE) - buffer,
        max(filtered_data$longitude, na.rm = TRUE) + buffer,
        max(filtered_data$latitude, na.rm = TRUE) + buffer
      )
    } else {
      leaflet() %>%
        addTiles() %>%
        setMaxBounds(

```

```

    lng1 = -180, lat1 = -60,
    lng2 = 180, lat2 = 85
  )
}
}
})
}

```

## Explanation

- `renderLeaflet`: dynamically generates interactive maps based on user selections.
- `addHeatmap`: displays data density across all institutions for the “All” option.
- `addMarkers`: shows markers for individual institutions, including their details in a popup.
- Dynamic Map Bounds:
  - `fitBounds`: Ensures the map zooms to fit the selected state or territory data.
  - Adds padding (buffer) for better visualization.

## Full code

```

# Install necessary packages (uncomment the line below if needed)
# install.packages(c("collegeScorecard", "leaflet", "leaflet.extras", "dplyr", "shiny"))

# Load required libraries
library(collegeScorecard) # Dataset containing information about U.S. colleges
library(leaflet) # For interactive maps
library(leaflet.extras) # Additional features for leaflet maps
library(dplyr) # Data manipulation
library(shiny) # Web application framework for R

# Load College Scorecard dataset
data("school")

# Create a mapping of state abbreviations to full state names
state_names <- data.frame(
  state = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA", "HI", "ID", "IL", "IN", "IA", "I",
  full_name = c("Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado", "Connecticut", "D",
)

# Merge university data with full state names
college_data <- school %>%
  filter(!is.na(latitude) & !is.na(longitude) & !is.na(state)) %>% # Remove missing location values
  left_join(state_names, by = "state") # Merge based on state abbreviation

# Separate U.S. states and territories into different lists
us_states <- state_names %>%
  filter(!(full_name %in% c("Puerto Rico", "American Samoa", "Guam", "Northern Mariana Islands", "U.S.
  pull(full_name)

```

```

territories <- state_names %>%
  filter(full_name %in% c("Puerto Rico", "American Samoa", "Guam", "Northern Mariana Islands", "U.S. Vi.
  pull(full_name)

# Define User Interface (UI)
ui <- fluidPage(
  titlePanel("Interactive Postsecondary Institution Map"), # Page title

  # Dropdown menu positioned at the top
  fluidRow(
    column(12,
      selectInput(
        "state",
        "Select a State or Territory:",
        choices = list(
          "All" = "All", # Default option to show all locations
          "U.S. States" = us_states, # Group states separately
          "Territories" = territories # Group territories separately
        ),
        selected = "All"
      )
    )
  ),

  # Map centered below
  fluidRow(
    column(12,
      leafletOutput("map", height = "600px") # Adjust the height for better visualization
    )
  )
)

# Define Server Logic
server <- function(input, output, session) {
  output$map <- renderLeaflet({
    if (input$state == "All") {
      # Heatmap for all institutions, when user choose All in the menu
      heatmap_data <- college_data
      leaflet(data = heatmap_data) %>%
        addTiles() %>%
        addProviderTiles(providers$CartoDB.Positron) %>%
        setMaxBounds(
          lng1 = -180, lat1 = -60, # Southwest corner of bounds
          lng2 = 180, lat2 = 85 # Northeast corner of bounds
        ) %>%
        addHeatmap(
          lng = ~longitude,
          lat = ~latitude,
          blur = 20,
          max = 0.05,
          radius = 15
        ) %>%

```

```

addMarkers(
  lng = ~longitude,
  lat = ~latitude,
  popup = ~paste0(
    "<b>Institution:</b> ", name, "<br>",
    ifelse(!is.na(deg_highest),
      paste0("<b>Highest degree offered:</b> ", as.character(deg_highest), "<br>"),
      paste0("<b>Predominant degree offered:</b> ", as.character(deg_predominant), "<br>")
    ),
    "<b>Website:</b> <a href='", url, "' target='_blank'>", url, "</a><br>",
    "<b>Location:</b> ", city, "<br>",
    ifelse(full_name %in% us_states,
      paste0("<b>State:</b> ", full_name, " (", state, ")<br>"),
      paste0("<b>Territory:</b> ", full_name, " (", state, ")<br>")
    )
  ),
  clusterOptions = markerClusterOptions()
)
} else {
  # Map with individual markers for a selected state or territory
  filtered_data <- college_data %>% filter(full_name == input$state)
  buffer <- 0.1

  if (nrow(filtered_data) > 0) {
    leaflet(data = filtered_data) %>%
      addTiles() %>%
      addProviderTiles(providers$CartoDB.Positron) %>%
      setMaxBounds(
        lng1 = -180, lat1 = -60,
        lng2 = 180, lat2 = 85
      ) %>%
      addMarkers(
        ~longitude,
        ~latitude,
        popup = ~paste0(
          "<b>Institution:</b> ", name, "<br>",
          ifelse(!is.na(deg_highest),
            paste0("<b>Highest degree offered:</b> ", as.character(deg_highest), "<br>"),
            paste0("<b>Predominant degree offered:</b> ", as.character(deg_predominant), "<br>")
          ),
          "<b>Website:</b> <a href='", url, "' target='_blank'>", url, "</a><br>",
          "<b>Location:</b> ", city, "<br>",
          ifelse(full_name %in% us_states,
            paste0("<b>State:</b> ", full_name, " (", state, ")<br>"),
            paste0("<b>Territory:</b> ", full_name, " (", state, ")<br>")
          )
        )
      ) %>%
      fitBounds(
        min(filtered_data$longitude, na.rm = TRUE) - buffer,
        min(filtered_data$latitude, na.rm = TRUE) - buffer,
        max(filtered_data$longitude, na.rm = TRUE) + buffer,
        max(filtered_data$latitude, na.rm = TRUE) + buffer
      )
    }
  }
}

```

```

    )
  } else {
    leaflet() %>%
      addTiles() %>%
      setMaxBounds(
        lng1 = -180, lat1 = -60,
        lng2 = 180, lat2 = 85
      )
  }
}
})
}
# Run the Shiny app
shinyApp(ui, server)

```