app.R

pearl

2024-06-14

library(shiny)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(ggplot2)  
library(readr)  
library(here)

## here() starts at C:/Users/pearl/OneDrive/Desktop/intership u/UOB Benchmarking Research Data/UOB Benchmarking Data

library(tidyr)  
library(jsonlite)

##   
## Attaching package: 'jsonlite'

## The following object is masked from 'package:shiny':  
##   
## validate

library(stringr)  
library(knitr)  
library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(DT)

##   
## Attaching package: 'DT'

## The following objects are masked from 'package:shiny':  
##   
## dataTableOutput, renderDataTable

ui <- fluidPage(  
 titlePanel("University Benchmarking Dashboard"),  
 sidebarLayout(  
 sidebarPanel(  
 selectInput("selectedUniversities", "Select Universities",   
 choices = c("University of Bradford", "University of Leeds", "University of Manchester", "University of Sheffield"),  
 selected = c("University of Bradford", "University of Leeds", "University of Manchester", "University of Sheffield"),  
 multiple = TRUE)  
 ),  
 mainPanel(  
 tabsetPanel(  
 tabPanel("Total vs Recent Citations",   
 plotlyOutput("plotCitations"),   
 dataTableOutput("tableCitations")),  
 tabPanel("Altmetric Score Comparison",   
 plotlyOutput("plotAltmetric")),  
 tabPanel("Research Output in Fields",   
 plotlyOutput("plotFORLevel2"),  
 plotlyOutput("plotFORLevel4")),  
 tabPanel("SDG Contributions",   
 plotlyOutput("plotSDG"),  
 dataTableOutput("tableSDG")),  
 tabPanel("Benchmarking Universities",   
 dataTableOutput("tableBenchmark"))  
 )  
 )  
 )  
)  
  
server <- function(input, output) {  
   
 # Load data  
 total\_citations\_df <- readr::read\_csv(here("university\_of\_bradford\_total\_citations.csv"))  
 recent\_citations\_df <- readr::read\_csv(here("university\_of\_bradford\_recent\_citations.csv"))  
 altmetric\_median\_df <- readr::read\_csv(here("university\_of\_bradford\_altmetric\_median.csv"))  
 publications\_for\_df <- readr::read\_csv(here("university\_of\_bradford\_publications\_for.csv"))  
 publications\_sdg\_df <- readr::read\_csv(here("university\_of\_bradford\_publications\_sdg.csv"))  
   
 # Combine total\_citations\_df and recent\_citations\_df  
 combined\_citations\_df <- merge(total\_citations\_df, recent\_citations\_df, by = c("id", "name"))  
   
 # Output for Total vs Recent Citations  
 output$plotCitations <- renderPlotly({  
 selected\_universities <- combined\_citations\_df %>%   
 filter(name %in% input$selectedUniversities)  
   
 plot <- ggplot(selected\_universities, aes(x = name)) +   
 geom\_bar(aes(y = citations\_total, fill = "Total Citations"), stat = "identity", position = position\_dodge()) +   
 geom\_bar(aes(y = recent\_citations\_total, fill = "Recent Citations"), stat = "identity", position = position\_dodge()) +   
 labs(title = "Total vs Recent Citations Comparison", x = "University", y = "Citations") +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +   
 scale\_fill\_manual(values = c("Total Citations" = "blue", "Recent Citations" = "red"))  
   
 ggplotly(plot)  
 })  
   
 output$plotAltmetric <- renderPlotly({  
 selected\_universities <- altmetric\_median\_df %>%   
 filter(name %in% input$selectedUniversities)  
   
 plot <- ggplot(selected\_universities, aes(x = name, y = altmetric\_median, fill = name)) +   
 geom\_bar(stat = "identity") +   
 labs(title = "Altmetric Score Comparison", x = "University", y = "Median Altmetric Score") +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +   
 scale\_fill\_brewer(palette = "Set1")  
   
 ggplotly(plot)  
 })  
   
 output$plotFORLevel2 <- renderPlotly({  
 publications\_for\_level2 <- publications\_for\_df %>% filter(level == 2)  
 plot <- ggplot(publications\_for\_level2, aes(x = reorder(name, -count), y = count, fill = name)) +   
 geom\_bar(stat = "identity") +   
 labs(title = "Research Output in FOR Level 2 Fields", x = "Field of Research", y = "Number of Publications") +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +   
 scale\_fill\_viridis\_d()  
   
 ggplotly(plot)  
 })  
   
 output$plotFORLevel4 <- renderPlotly({  
 publications\_for\_level4 <- publications\_for\_df %>% filter(level == 4)  
 plot <- ggplot(publications\_for\_level4, aes(x = reorder(name, -count), y = count, fill = name)) +   
 geom\_bar(stat = "identity") +   
 labs(title = "Research Output in FOR Level 4 Fields", x = "Field of Research", y = "Number of Publications") +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +   
 scale\_fill\_viridis\_d()  
   
 ggplotly(plot)  
 })  
   
 output$plotSDG <- renderPlotly({  
 sdg\_summary <- publications\_sdg\_df %>%   
 arrange(desc(count))  
   
 plot <- ggplot(sdg\_summary, aes(x = reorder(name, -count), y = count, fill = name)) +   
 geom\_bar(stat = "identity") +   
 labs(title = "Overall Contribution to Sustainable Development Goals", x = "SDG", y = "Number of Publications") +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +   
 scale\_fill\_viridis\_d()  
   
 ggplotly(plot)  
 })  
   
 output$tableSDG <- DT::renderDataTable({  
 sdg\_summary <- publications\_sdg\_df %>%   
 arrange(desc(count))  
 DT::datatable(sdg\_summary)  
 })  
   
 output$tableBenchmark <- DT::renderDataTable({  
 selected\_universities <- input$selectedUniversities  
   
 comparison\_df <- total\_citations\_df %>%   
 filter(name %in% selected\_universities) %>%   
 select(name, citations\_total) %>%   
 left\_join(recent\_citations\_df %>% select(name, recent\_citations\_total), by = "name") %>%   
 left\_join(altmetric\_median\_df %>% select(name, altmetric\_median), by = "name") %>%   
 left\_join(publications\_for\_df %>% select(name, count), by = "name") %>%   
 left\_join(publications\_sdg\_df %>% select(name, count), by = "name")  
   
 DT::datatable(comparison\_df)  
 })  
}  
  
# Run the application   
shinyApp(ui = ui, server = server)