Zachary Villa
DTSC 450
05/05/2025

app.R Code

```
# Load packages
library(shiny)
library(bslib)
library(ggplot2)
library(dplyr)
library(DT)
library(plotly)

# Loading data
team_statistics <- read.csv("team_statistics.csv")
roster_info <- read.csv("roster_info.csv")

# Define UI for application
ui <- navbarPage(
  title = "March Madness",

  # "Compare Teams" tab
  # Setting up page format
  tabPanel("Compare Teams",
           fluidPage(
             h1("Compare Teams"),
             sidebarLayout(
               sidebarPanel(
                 #Gathering user input
                 selectInput("compare_year", "Select Year:",
                             choices = unique(team_statistics$YEAR)),
                selectInput("compare_stat", "Select Stat to Compare:",
                             choices = c("Wins" = "W",
                                         "Losses" = "L",
                                         "Win%" = "WINPCT",
                                         "ADJOE" = "ADJOE",
                                         "ADJDE" = "ADJDE",
                                         "EFG%" = "EFG_O",
                                         "Opp. EFG%" = "EFG_D",
```

```
                                "TOR" = "TOR",
                                "Opp. TOR" = "TORD",
                                "ORB" = "ORB",
                                "DRB" = "DRB",
                                "FTR" = "FTR",
                                "Opp. FTR" = "FTRD",
                                "2P%" = "X2P_O",
                                "Opp. 2P%" = "X2P_D",
                                "3P%" = "X3P_O",
                                "Opp. 3P%" = "X3P_D",
                                "ADJT" = "ADJ_T",
                                "Wins Above Bubble" = "WAB",
                                "Seed" = "SEED")),
            selectizeInput("compare_team", "Select (up to 4) Teams:",
                        choices = unique(team_statistics$TEAM),
                        multiple = TRUE,
                        options = list(maxItems = 4))
          ),

          mainPanel(
            #Displaying bar chart in main panel
            plotOutput("compare_bar_chart")
          )
        )
       )),

# "Team Information" tab
# Setting up page format
tabPanel("Team Information",
         fluidPage(
          h1("Team Information"),
          sidebarLayout(
            sidebarPanel(
              #Gathering user input
              selectInput("information_year", "Select Year:",
                          choices = unique(team_statistics$YEAR)),
              selectInput("information_team", "Select Team:",
                          choices = unique(roster_info$Team))
            ),
            mainPanel(
```

```
            # Displaying data tables in main panel
            tableOutput("team_table"),
            tableOutput("roster_table")
          )
        )
      )),

  # "Team Placement" tab
  # Setting up page format
  tabPanel("Tournament Placement",
      fluidPage(
        h1("Tournament History"),
        sidebarLayout(
          sidebarPanel(
            # Gathering user input
            selectInput('placement_year', "Select Year:",
                  choices = unique(team_statistics$YEAR)),
            selectInput('placement', "Select Placement:",
                  choices = c('First Four',
                        'First Round',
                        'Second Round',
                        'Sweet Sixteen',
                        'Elite Eight',
                        'Final Four',
                        'Runner-up',
                        'Champion'))
          ),
          mainPanel(
            # Displaying data table in main panel
            tableOutput("placement_table")
          )
        )
      )),

  # "About" tab
  tabPanel("About",
      fluidPage(
        p("Every March, college basketball fans are treated to one of the most thrilling events in
sports: March Madness.
```

```
            This Shiny app provides an interactive way to explore NCAA men's basketball
tournament data — from team statistics to
            tournament placements — to help fans, analysts, and curious users better understand
the performance and trends of teams
            throughout the years."),
          p("This application includes four main sections:"),
          h3("Compare Teams"),
          p("Use this tab to compare the performance of up to four teams in a selected year across
various statistical categories.
           Choose from metrics like Win %, Offensive/Defensive Efficiency (ADJOE, ADJDE),
Rebounding (ORB, DRB), Shooting Percentages
            (2P%, 3P%, EFG%), Tempo (ADJT), Seed, and more. A bar chart dynamically updates
based on your selections, making it easy to
            visualize how the teams stack up in your chosen stat."),
          h3("Team Information"),
          p("This tab provides detailed information for any selected team and year. View the
team's full statistical profile for that season.
            See the complete roster, including player names and other available data. Perfect for
fans who want a deep dive into their favorite
            team's performance and personnel in a given year."),
          h3("Tournament History"),
          p("Curious about how teams have historically performed in the tournament? Select a
specific year and placement
            (e.g., Final Four, Runner-up, Champion) to see which teams reached that stage.Results
include seed, team name, record, and postseason
            finish, helping you explore trends like underdog runs and dominant performances.")
        )),
)

# Define server logic
server <- function(input, output, session) {

  # "Compare Teams" tab

  # Creating variable to store filtered data using user input
  filtered_compare <- reactive({
    filter(team_statistics, YEAR == input$compare_year, TEAM %in% input$compare_team)
  })
  # Formatting bar chart
  output$compare_bar_chart <- renderPlot ({
```

```
    ggplot(filtered_compare(), aes(x = TEAM, y = .data[[input$compare_stat]], fill = TEAM)) +
      geom_bar(stat = "identity") +
      labs(title = "Stat Comparison", x = "Team", y = input$compare_stat)
  })

  # "Team Information" tab

  # Filtering data using user input
  output$team_table <- renderTable({
    filter(team_statistics, YEAR == input$information_year, TEAM == input$information_team)
  })

  # Filtering data using user input
  output$roster_table <- renderTable({
      filter(roster_info, Year == input$information_year, Team == input$information_team)
  })

  # "Tournament Placement" tab

   # Filtering data using user input
  output$placement_table <- renderTable({
    filter(team_statistics, YEAR == input$placement_year, POSTSEASON == input$placement)
%>%
    # Only displaying SEED, TEAM, RECORD, & POSTSEASON in this data table
    select(SEED, TEAM, RECORD, POSTSEASON)
  })
}

# Link UI and server and run the application
shinyApp(ui = ui, server = server)
```

Zachary Villa
DTSC 450
05/05/2025

team_statistics.Rmd Code

Loading packages
```{r}
library(tidyverse)
library(dplyr)
library(tidyr)
```

Loading dataset
```{r}
team_statistics <- read.csv("cbb.csv")
```

Previewing dataset
```{r}
head(team_statistics)
```

Getting rid of BARTHAG variable
```{r}
team_statistics$BARTHAG <- NULL
```

Defining the variable SEED as an integer
```{r}
team_statistics$SEED <- as.integer(team_statistics$SEED)
```

Renaming the placements of the POSTSEASON variable
```{r}
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "Champions", "Champion")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "2ND", "Runner-up")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "F4", "Final Four")
```

```
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "E8", "Elite Eight")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "S16", "Sweet Sixteen")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "R32", "Second Round")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "R64", "First Round")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "R68", "First Four")
team_statistics$POSTSEASON <- replace(team_statistics$POSTSEASON,
team_statistics$POSTSEASON  == "N/A", NA)
```

Filtering dataset down to only teams that made the tournament
```{r}
team_statistics <- filter(team_statistics, !is.na(POSTSEASON))
```

Creating L variable
```{r}
team_statistics <- mutate(team_statistics, L = G - W)
```

Creating RECORD variable
```{r}
team_statistics <- mutate(team_statistics, RECORD = paste(W, "-", L))

```

Creating WINPCT variable
```{r}
team_statistics <- mutate(team_statistics, WINPCT = W / G)
team_statistics <- mutate(team_statistics, WINPCT = round(WINPCT, 3))
```

Reordering columns
```{r}
team_statistics <- relocate(team_statistics, L, .after = W)
team_statistics <- relocate(team_statistics, RECORD, .after = G)
```

```
team_statistics <- relocate(team_statistics, WINPCT, .after = L)
```

Arranging columns by TEAM and YEAR
```{r}
team_statistics <- arrange(team_statistics, YEAR, TEAM)
```

Viewing final table
```{r}
view(team_statistics)
```

Zachary Villa
DTSC 450
05/05/2025

Documentation

- Tutorial used to learn Shiny: click here

- Page Layout
  - navbarPage
  - tabPanel
  - fluidPage
  - sidebarLayout, sidebarPanel, & mainPanel

- selectizeInput("compare_team", "Select (up to 4) Teams:",
  choices = unique(team_statistics$TEAM),
  multiple = TRUE,
  options = list(maxItems = 4))

  Reference: click here (Lines 85 -90)

- Allow user to select input
  - selectInput & selectizeInput