## URL to dashboard

## <https://bianca-bosco.shinyapps.io/FIFAWomen/>

## Code

library(shiny)

library(ggplot2)

library(dplyr)

library(tidyr)

# Data loading and preparation

world\_cup\_data <- data.frame(

Year = c(2023, 2019, 2015, 2011, 2007, 2003, 1999, 1995, 1991),

Host = c("Australia, New Zealand", "France", "Canada", "Germany", "China PR",

"United States", "United States", "Sweden", "China PR"),

Teams = c(32, 24, 24, 16, 16, 16, 16, 12, 12),

Champion = c("Spain", "United States", "United States", "Japan", "Germany",

"Germany", "United States", "Norway", "United States"),

Runner\_Up = c("England", "Netherlands", "Japan", "United States", "Brazil",

"Sweden", "China PR", "Germany", "Norway"),

Third\_Place = c("Sweden", "Sweden", "England", "Sweden", "United States",

"United States", "Brazil", "United States", "Sweden")

)

df2 <- read.csv("womens-world-cup.csv")

# Checking if columns exist

if(!all(c("year", "squad", "goals", "assists") %in% colnames(df2))) {

stop("The dataset df2 must contain the columns: year, squad, goals, assists.")

}

# Prepare data for player goals

player\_goals <- data.frame(

Year = c(1991, 1995, 1999, 2003, 2007, 2011, 2015, 2019, 2023),

Marta = c(NA, NA, NA, 3, 7, 4, 1, 2, 0),

Birgit\_Prinz = c(NA, 1, 1, 7, 5, 0, NA, NA, NA),

Abby\_Wambach = c(NA, NA, NA, 3, 6, 4, 1, NA, NA),

Michelle\_Akers = c(10, 0, 2, NA, NA, NA, NA, NA, NA),

Sun\_Wen = c(1, 2, 7, 1, NA, NA, NA, NA, NA),

Cristiane = c(NA, NA, NA, 0, 5, 2, 0, 4, NA),

Bettina\_Wiegmann = c(3, 3, 3, 2, NA, NA, NA, NA, NA)

)

# Pivot for ggplot

data\_long <- player\_goals %>%

pivot\_longer(cols = -Year, names\_to = "Player", values\_to = "Goals") %>%

filter(!is.na(Goals))

# define UI

ui <- fluidPage(

titlePanel("FIFA Women's World Cup Statistics"),

sidebarLayout(

sidebarPanel(

sliderInput("yearInput", "Select Year:",

min = min(world\_cup\_data$Year),

max = max(world\_cup\_data$Year),

value = min(world\_cup\_data$Year),

step = 4,

animate = TRUE),

selectInput("playerInput", "Select Player:",

choices = colnames(player\_goals)[-1])

),

mainPanel(

tabsetPanel(

tabPanel("Main",

fluidRow(

column(6, plotOutput("teamsPlot", height = "300px")),

column(6, plotOutput("barChartPlot", height = "300px"))

),

fluidRow(

htmlOutput("yearInfo")

),

fluidRow(

tabsetPanel(

tabPanel("Goals Scored",

fluidRow(

column(6, plotOutput("playerGoalsPlot", height = "300px")),

column(6, plotOutput("totalGoalsPlot", height = "300px"))

)),

tabPanel("No of Medals",

plotOutput("medalPlot", height = "300px"))

)

)

),

tabPanel("References",

tags$h3("References"),

tags$ul(

tags$li("FIFA Women’s World Cup Stats. (www.sports-reference.com/). Www.kaggle.com. https://www.kaggle.com/datasets/mattop/fifa-womens-world-cup-stats"),

tags$li("FIFA. (2023, December 9). FIFA Women’s World Cup all-time leading scorers. FIFA. https://www.fifa.com/en/tournaments/womens/womensworldcup/articles/womens-world-cup-all-time-leading-scorers-goal-top-list"),

tags$li("Wikipedia Contributors. (2020, August 8). FIFA Women’s World Cup records and statistics. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/FIFA\_Women%27s\_World\_Cup\_records\_and\_statistics")

)

)

)

)

)

)

# define server

server <- function(input, output) {

selected\_year <- reactive({

world\_cup\_data[world\_cup\_data$Year == input$yearInput, ]

})

output$teamsPlot <- renderPlot({

year\_data <- selected\_year()

ggplot(world\_cup\_data, aes(x = Year, y = Teams)) +

geom\_line() +

geom\_point() +

geom\_vline(xintercept = input$yearInput, linetype = "dashed", color = "red") +

labs(title = "Number of Teams Over the Years",

x = "Year",

y = "Number of Teams")

})

output$yearInfo <- renderUI({

year\_data <- selected\_year()

HTML(paste(

"<b>Host:</b>", year\_data$Host,

"<b>Champion:</b>", year\_data$Champion,

"<b>Runner-Up:</b>", year\_data$Runner\_Up,

"<b>Third Place:</b>", year\_data$Third\_Place

))

})

output$medalPlot <- renderPlot({

medal\_count <- world\_cup\_data %>%

pivot\_longer(cols = c(Champion, Runner\_Up, Third\_Place), names\_to = "Medal\_Type", values\_to = "Country") %>%

group\_by(Country, Medal\_Type) %>%

summarize(Total\_Medals = n(), .groups = 'drop') %>%

mutate(Medal\_Type = factor(Medal\_Type, levels = c("Champion", "Runner\_Up", "Third\_Place"))) %>%

arrange(desc(Total\_Medals))

ggplot(medal\_count, aes(x = Total\_Medals, y = reorder(Country, -Total\_Medals), fill = Medal\_Type)) +

geom\_bar(stat = "identity") +

geom\_text(aes(label = Total\_Medals), position = position\_stack(vjust = 0.5)) +

labs(title = "No of Medals by Country",

x = "No of Medals",

y = "Country") +

scale\_fill\_manual(values = c("Champion" = "#FFD700", "Runner\_Up" = "#C0C0C0", "Third\_Place" = "#CD7F32"),

name = "Medal",

labels = c("Gold", "Silver", "Bronze")) +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 0, hjust = 0.5),

plot.title = element\_text(hjust = 0.5),

legend.position = "top",

legend.title = element\_text(face = "bold"),

panel.grid.major.y = element\_blank(),

panel.grid.minor.y = element\_blank()) +

guides(fill = guide\_legend(reverse = TRUE))

})

output$barChartPlot <- renderPlot({

data <- df2 %>% filter(year == input$yearInput) %>%

arrange(desc(goals)) %>%

top\_n(10, goals) %>%

gather(key = "Metric", value = "Value", goals, assists)

ggplot(data, aes(x = reorder(squad, -Value), y = Value, fill = Metric)) +

geom\_bar(stat = "identity", position = "dodge", width = 0.9) +

geom\_text(aes(label = ifelse(Value == 0, "", Value)),

position = position\_dodge(width = 0.9),

vjust = 0.5, hjust = ifelse(data$Metric == "goals", -0.25, -0.25)) +

coord\_flip() +

scale\_fill\_manual(values = c("goals" = "#1f78b4", "assists" = "#33a02c"), name = "Metric") +

labs(title = paste("Top 10 Countries by Goals and Assists in", input$yearInput),

x = "Country",

y = "Count") +

theme\_minimal() +

theme(plot.title = element\_text(hjust = 0.5))

})

output$playerGoalsPlot <- renderPlot({

player\_name <- input$playerInput

player\_data <- data\_long %>% filter(Player == player\_name)

ggplot(player\_data, aes(x = Year, y = Goals, fill = Player)) +

geom\_bar(stat = "identity") +

labs(title = paste("Goals Scored by", player\_name, "-", ifelse(player\_name == "Marta", "Brazil",

ifelse(player\_name == "Birgit\_Prinz", "Germany",

ifelse(player\_name == "Abby\_Wambach", "USA",

ifelse(player\_name == "Michelle\_Akers", "USA",

ifelse(player\_name == "Sun\_Wen", "China PR",

ifelse(player\_name == "Cristiane", "Brazil", "Germany"))))))),

x = "Year",

y = "Number of Goals") +

scale\_x\_continuous(breaks = seq(1991, 2023, by = 4)) +

theme\_minimal() +

theme(plot.title = element\_text(hjust = 0.5))

})

output$totalGoalsPlot <- renderPlot({

total\_goals <- data\_long %>%

group\_by(Player) %>%

summarize(Total\_Goals = sum(Goals, na.rm = TRUE)) %>%

mutate(is\_selected = Player == input$playerInput)

ggplot(total\_goals, aes(x = reorder(Player, -Total\_Goals), y = Total\_Goals, fill = is\_selected)) +

geom\_bar(stat = "identity") +

geom\_text(aes(label = Total\_Goals), vjust = -0.3) +

scale\_fill\_manual(values = c("TRUE" = "#FF9999", "FALSE" = "#999999"), guide = "none") +

labs(title = "Total Goals Scored by Top Players",

x = "Player",

y = "Total Goals") +

theme\_minimal() +

theme(plot.title = element\_text(hjust = 0.5),

axis.text.x = element\_text(angle = 45, hjust = 1))

})

}

# Run the app

shinyApp(ui = ui, server = server)

## References

[1] FIFA Women’s World Cup Stats. (www.sports-reference.com/). Www.kaggle.com. https://www.kaggle.com/datasets/mattop/fifa-womens-world-cup-stats

[2] FIFA. (2023, December 9). FIFA Women’s World Cup all-time leading scorers. FIFA. https://www.fifa.com/en/tournaments/womens/womensworldcup/articles/womens-world-cup-all-time-leading-scorers-goal-top-list

[3] Wikipedia Contributors. (2020, August 8). FIFA Women’s World Cup records and statistics. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/FIFA\_Women%27s\_World\_Cup\_records\_and\_statistics

[4] Rawat, H. (2023, December 4). Summer Olympic Insights: Analyzing Games Data with SQL & Power BI. Medium. https://medium.com/@himanshu\_rawat2003/summer-olympic-insights-analyzing-games-data-with-sql-power-bi-32e24092b360

[5] Baglin, J. (2023). Chapter 10. Dashboards. Retrieved from <https://dark-star-161610.appspot.com/secured/_book/dashboards.html#conclusion-1>

[6] R Shiny in Life Sciences - Top 7 Dashboard Examples. (n.d.). Www.appsilon.com. Retrieved June 17, 2024, from https://www.appsilon.com/post/r-shiny-in-life-sciences-examples