

RShiny - Employee Demographic (Markdown)

Laskow

2023-12-07

```
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(rsconnect)
```

```
##
## Attaching package: 'rsconnect'
## The following object is masked from 'package:shiny':
##
##   serverInfo
```

```
library(RCurl)
library(aws.s3)
```

```
attritiondata_original<-read.table(textConnection(getURL(
  "https://s3.us-east-2.amazonaws.com/msds.ds.6306.2/CaseStudy2-data.csv"
)), sep="," , header=TRUE)
```

```
data<-data.frame(attritiondata_original)
```

```
# UI
```

```

ui <- fluidPage(
  titlePanel("Distribution of Employee Age"),
  sidebarLayout(
    sidebarPanel(
      selectInput(inputId = "Department",
        label = "Department",
        choices = unique(data$Department)),
      hr(),
      helpText("Age distribution of Employees by Department.")
    ),
    mainPanel(
      plotOutput("AgePlot"),
      tableOutput("mytable")
    )
  )
)

server <- function(input, output){
  output$AgePlot <- renderPlot({
    # Filter data by the selected department
    department_data <- data[data$Department == input$Department, ]

    # Plotting the bar plot
    barplot(table(department_data$Age),
      main = input$Department,
      ylab = "Count",
      xlab = "Age")
  })

  output$summary <- renderPrint({
    department_data <- data[data$Department == input$Department, "Age"]
    total_count <- nrow(department_data)
    summary_stats <- summary(department_data)

    cat("Summary for", input$Department, "\n")
    cat("Total Count:", total_count, "\n")
    print(summary_stats)
  })

  output$mytable <- renderTable({
    department_data <- data[data$Department == input$Department, "Age"]
    summary_stats <- data.frame(
      "Total Count" = length(department_data),
      "Mean Age" = mean(department_data, na.rm = TRUE),
      "Median Age" = median(department_data, na.rm = TRUE),
      "STDEV Age" = sd(department_data, na.rm = TRUE),
      "Min Age" = min(department_data, na.rm = TRUE),
      "Max Age" = max(department_data, na.rm = TRUE)
    )
  })
}

```

```
shinyApp(ui = ui, server = server)
```