app.R

brett.johnson

Mon Jul 16 12:55:18 2018

```
library(shiny)
```

```
## Warning: package 'shiny' was built under R version 3.4.4
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## 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)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
survey_seines <- readRDS("data/survey_seines.RDS")
spp_labels <- c(CU = "Chum", PI = "Pink", SO = "Sockeye", DI = "Discovery Islands",
                JS = "Johnstone Strait")

ui <- navbarPage(
  title = 'Hakai Institute Juvenile Salmon Program',
  tabPanel("Overview",
           sidebarLayout(
             sidebarPanel(
               helpText(),
               downloadButton("report", "Generate PDF report")
             ),
             mainPanel(
               includeMarkdown('overview.md'),
               img(src = "map_2018.jpg", height = 800, width = 960)
             )
           )
  ),
  tabPanel("Migration Timing",
           sidebarLayout(
             sidebarPanel(
               helpText(),
               selectInput("species", label = h3("Species"),
                           choices = list("Sockeye" = "so_total", "Pink" = "pi_total",
                                          "Chum" = "cu_total"),
                           selected = "Sockeye")
             ),
             mainPanel(
               plotOutput("migration_timing")
             )
           )
  )
)

# Define server logic required to draw a histogram
server <- function(input, output) {
  output$report <- downloadHandler(
    # For PDF output, change this to "report.pdf"
    filename = "report",
    content = function(file) {
      # Copy the report file to a temporary directory before processing it, in
      # case we don't have write permissions to the current working dir (which
      # can happen when deployed).
      tempReport <- file.path(tempdir(), "report.Rmd")
      file.copy("report.Rmd", tempReport, overwrite = TRUE)

      # Set up parameters to pass to Rmd document
      params <- list(n = input$slider)

      # Knit the document, passing in the `params` list, and eval it in a
      # child of the global environment (this isolates the code in the document
      # from the code in this app).
      rmarkdown::render(tempReport, output_file = file,
                        params = params,
                        envir = new.env(parent = globalenv())
      )
    }
  )
  output$migration_timing <- renderPlot({
    species_selected <- input$species

    survey_seines %>%
      select(year, region, sampling_week, species_selected) %>%
      group_by(year, region, sampling_week) %>%
      summarise(mean = mean(get(input$species), na.rm = T), se = sd(get(input$species)) / sqrt(n())) %>
      ungroup() %>%
      ggplot(aes(x = as_factor(sampling_week), y = mean, colour = region, group = region))+
      geom_line(size = 1)+
      geom_point() +
      geom_errorbar(aes(ymin = mean - se, ymax = mean + se, width = 0.2)) +
      scale_colour_discrete(name = "",
                            breaks=c("DI","JS"),
                            labels=c("Discovery Islands", "Johnstone Strait"))+
      theme(legend.justification=c(1,0), legend.position=c(.8,.8),
            legend.background = element_rect(fill=alpha(0.1))) +
      theme(legend.text = element_text(colour="black", size = 12)) +
      theme(axis.text=element_text(size=12),
            axis.title=element_text(size=12,face="bold")) +
      xlab("Date") +
      ylab("Abundance") +
```

```r
      theme(legend.title = element_blank()) +
      theme(axis.text.x = element_text(angle = 45, vjust = 0.5))

  }
  )
}

# Run the application
shinyApp(ui = ui, server = server)
```

Shiny applications not supported in static R Markdown documents