MATH2270/MATH2237/MATH2404 Assignment 3

Name
Eran Dodampe Gamage

Student number
s4074647

Assessment declaration checklist
Please carefully read the statements below and check each box if you agree with the declaration. If you do not check all boxes, your assignment will not be marked. If you make a false declaration on any of these points, you may be investigated for academic misconduct. Students found to have breached academic integrity may receive official warnings and/or serious academic penalties. Please read more about academic integrity here. If you are unsure about any of these points or feel your assessment might breach academic integrity, please contact your course coordinator for support. It is important that you DO NOT submit any assessment until you can complete the declaration truthfully.

By checking the boxes below, I declare the following:

- ☒ I have not impersonated, or allowed myself to be impersonated by, any person for the purposes of this assessment
- ☒ This assessment is my original work and no part of it has been copied from any other source except where due acknowledgement is made.
- ☒ No part of this assessment has been written for me by any other person except where such collaboration has been authorised by the lecturer/teacher concerned.
- ☒ I have not used generative “AI” tools for the purposes of this assessment.
- ☒ Where this work is being submitted for individual assessment, I declare that it is my original work and that no part has been contributed by, produced by or in conjunction with another student.
- ☒ I give permission for my assessment response to be reproduced, communicated, compared and archived for the purposes of detecting plagiarism.
- ☒ I give permission for a copy of my assessment to be retained by the university for review and comparison, including review by external examiners.

I understand that:

- ☒ Plagiarism is the presentation of the work, idea or creation of another person or machine as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
- ☒ Plagiarism includes the act of assisting or allowing another person to plagiarise or to copy my work.

I agree and acknowledge that:

- ☒ I have read and understood the Declaration and Statement of Authorship above.

☒ If I do not agree to the Declaration and Statement of Authorship in this context and all boxes are not checked, the assessment outcome is not valid for assessment purposes and will not be included in my final result for this course.

Assignment URL

https://0a38og-eran-dodampe0gamage.shinyapps.io/assessment_3_storytelling_with_open_data/

References

Ritchie, H., Rosado, P., & Roser, M. (2024). *Energy data*. Our World in Data. https://github.com/owid/energy-data

Natural Earth. (2023). *World countries GeoJSON* [Geographic data file]. Natural Earth. https://www.naturalearthdata.com/

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2023). *shiny: Web application framework for R* (Version 1.8.0) [Computer software]. https://CRAN.R-project.org/package=shiny

Chang, W., & Borges Ribeiro, B. (2021). *shinydashboard: Create dashboards with 'Shiny'* (Version 0.7.2) [Computer software]. https://CRAN.R-project.org/package=shinydashboard

Sievert, C. (2020). *Interactive web-based data visualization with R, plotly, and shiny*. Chapman and Hall/CRC. https://plotly-r.com

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A grammar of data manipulation* (Version 1.1.4) [Computer software]. https://CRAN.R-project.org/package=dplyr

Wickham, H., Hester, J., & Bryan, J. (2023). *readr: Read rectangular text data* (Version 2.1.4) [Computer software]. https://CRAN.R-project.org/package=readr

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*, *10*(1), 439-446. https://doi.org/10.32614/RJ-2018-009

Wickham, H., & Seidel, D. (2022). *scales: Scale functions for visualization* (Version 1.3.0) [Computer software]. https://CRAN.R-project.org/package=scales

Wickham, H., Vaughan, D., & Girlich, M. (2023). *tidyr: Tidy messy data* (Version 1.3.0) [Computer software]. https://CRAN.R-project.org/package=tidyr

jQuery Foundation. (2023). *jQuery* (Version 3.6.0) [JavaScript library]. https://jquery.com/

Agafonkin, V. (2023). *Leaflet* (Version 1.9.4) [JavaScript library]. https://leafletjs.com/

Plotly Technologies Inc. (2023). *Plotly.js* (Version 2.26.0) [JavaScript library]. https://plot.ly/javascript/

Font Awesome. (2023). *Font Awesome* (Version 6.4.0) [Icon library]. https://fontawesome.com/

Bootstrap contributors. (2023). *Bootstrap* (Version 3.4.1) [CSS framework]. https://getbootstrap.com/

Harrower, M., & Brewer, C. A. (2003). ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, *40*(1), 27-37. https://doi.org/10.1179/000870403235002042

International Energy Agency. (2023). *Net zero by 2050: A roadmap for the global energy sector*. IEA. https://www.iea.org/reports/net-zero-by-2050

Intergovernmental Panel on Climate Change. (2023). *Climate change 2023: Synthesis report*. IPCC. https://www.ipcc.ch/report/ar6/syr/

United Nations Framework Convention on Climate Change. (2015). *Paris Agreement*. United Nations. https://unfccc.int/process-and-meetings/the-paris-agreement

RStudio. (2023). *Shiny dashboard documentation*. https://rstudio.github.io/shinydashboard/

Sievert, C. (2020). *Interactive web-based data visualization with R, plotly, and shiny*. Chapman and Hall/CRC. https://plotly-r.com

Assignment Code

app.R

```
# Global Energy Transition Dashboard

# Author: Eran Dodampe Gamage

# Data Source: Our World in Data


library(shiny)

library(shinydashboard)

library(plotly)

library(dplyr)

library(readr)

library(sf)

library(scales)

library(tidyr)
```

```
# Data Preparation

energy_raw <- read_csv("data/owid-energy-data.csv", show_col_types = FALSE)
world <- st_read("data/world-countries.json", quiet = TRUE)

region_map <- world %>%
  st_drop_geometry() %>%
  select(iso = iso_a3, region = region_un)

energy <- energy_raw %>%
  select(
    country, iso_code, year,
    renewables_share_elec, renewables_electricity,
    fossil_electricity, nuclear_electricity,
    energy_per_capita, greenhouse_gas_emissions,
    gdp, population
  ) %>%
  left_join(region_map, by = c("iso_code" = "iso")) %>%
  mutate(
    total_electricity = fossil_electricity + renewables_electricity +
      coalesce(nuclear_electricity, 0),
    co2_intensity = case_when(
      total_electricity > 0 ~ greenhouse_gas_emissions * 1e6 / total_electricity,
      TRUE ~ NA_real_
    ),
    renewables_share_elec = pmin(renewables_share_elec, 100)
  ) %>%
  filter(!is.na(renewables_share_elec), !is.na(region), year >= 2000)
```

```
# Unser Interface

ui <- dashboardPage(
  title = "Global Energy Transition",
  skin = "black",

  dashboardHeader(
    title = "Global Energy Transition Dashboard",
    titleWidth = 250
  ),

  dashboardSidebar(
    width = 250,
    sidebarMenu(
      id = "tabs",
      menuItem("Current State", tabName = "current", icon = icon("globe-americas")),
      menuItem("Evolution", tabName = "evolution", icon = icon("chart-line")),
      menuItem("Impact Analysis", tabName = "impact", icon = icon("chart-area"))
    ),

    div(class = "dashboard-description",
        p("Explore the global transition to renewable energy sources and their impact on carbon emissions from 2000 to present.")
    ),

    div(class = "year-container",
        h5("Select Year"),
        sliderInput("year", NULL,
```

```
                min = 2000,
                max = max(energy$year),
                value = max(energy$year),
                sep = "",
                animate = FALSE,
                width = "100%",
                ticks = TRUE),
    actionButton("playBtn",
                 label = "",
                 icon = icon("play"),
                 class = "btn-play-slider")
  ),

  div(class = "metric-container global-average",
    h5("Global Average"),
    div(class = "metric-value", textOutput("globalAvgText")),
    div(class = "metric-label", "Renewable Energy")
  ),

  div(class = "sidebar-footer",
    p("Data: Our World in Data"),
    p("Dashboard: Eran Dodampe Gamage")
  )
),

dashboardBody(
  tags$head(
    tags$link(rel = "stylesheet", type = "text/css", href = "style.css"),
    tags$script(HTML("
```

```
$(document).ready(function() {
  $('body').css('overflow', 'hidden');

  // Play button functionality
  var playing = false;
  var interval;

  $(document).on('click', '#playBtn', function(e) {
    e.preventDefault();

    if (!playing) {
      playing = true;
      $(this).find('i').removeClass('fa-play').addClass('fa-pause');

      // Get slider instance
      var slider = $('#year').data('ionRangeSlider');
      if (!slider) return;

      var currentVal = slider.result.from;
      var maxVal = slider.result.max;
      var minVal = slider.result.min;

      // Reset to start if at end
      if (currentVal >= maxVal) {
        slider.update({from: minVal});
      }

      // Animate through years
      interval = setInterval(function() {
```

```
            var current = slider.result.from;
            if (current < maxVal) {
              slider.update({from: current + 1});
            } else {
              clearInterval(interval);
              playing = false;
              $('#playBtn').find('i').removeClass('fa-pause').addClass('fa-play');
            }
          }, 2000);
        } else {
          // Pause
          playing = false;
          clearInterval(interval);
          $(this).find('i').removeClass('fa-pause').addClass('fa-play');
        }
      });
    });
  ")),
  tags$style(HTML("
    * {
      font-family: 'Gill Sans', sans-serif !important;
    }
  "))
),

tabItems(
  # Current State Tab
  tabItem(
    tabName = "current",
```

```
    fluidRow(
      column(12,
        div(class = "section-header",
          h3("Global Renewable Energy Distribution"),
          div(class = "section-subtitle",
            "Percentage of electricity generated from renewable sources by country"
          ),
          div(class = "section-description",
            "This map visualizes the current state of renewable energy adoption worldwide. Countries with higher renewable shares appear in darker green, indicating greater progress in the energy transition.")
        )
      )
    ),
    fluidRow(
      style = "margin-bottom: 30px;",
      column(12,
        div(class = "main-viz-container",
          plotlyOutput("worldMap", height = "600px")
        )
      )
    ),
    fluidRow(
      column(4,
        div(class = "stat-box",
          div(class = "stat-value", textOutput("statCountries")),
          div(class = "stat-label", "Countries Tracked")
        )
      ),
      column(4,
```

```
            div(class = "stat-box",
                div(class = "stat-value", textOutput("statLeaders")),
                div(class = "stat-label", "Above 50% Renewable")
            )
        ),
        column(4,
            div(class = "stat-box",
                div(class = "stat-value", textOutput("statGrowth")),
                div(class = "stat-label", "Year-over-Year Change")
            )
        )
      )
    ),

    # Evolution Tab
    tabItem(
      tabName = "evolution",
      fluidRow(
        style = "margin-bottom: 15px;",
        column(12,
            div(class = "section-header",
                h3("Energy Transition Timeline"),
                p("Tracking renewable energy adoption in major economies from 2000 to present"),
                p(class = "section-description",
                  "This timeline reveals how different countries have progressed in their renewable energy journey. Key policy events marked below have significantly influenced global energy transitions.")
            )
        )
      ),
```

```
    fluidRow(
      style = "margin-bottom: 15px;",
      column(8,
        div(class = "viz-container",
          plotlyOutput("timeSeriesPlot", height = "350px")
        )
      ),
      column(4,
        div(class = "viz-container compact",
          h4("Progress Since 2000"),
          plotlyOutput("progressPlot", height = "200px")
        )
      )
    ),
    fluidRow(
      column(12,
        div(class = "events-strip",
          h4("Key Global Energy Policy Events"),
          div(class = "events-grid",
            div(class = "event-card",
              div(class = "event-year-card", "2011"),
              div(class = "event-title-card", "Fukushima Disaster"),
              div(class = "event-desc-card", "Nuclear disaster led to decisions to phase out nuclear by 2022. Massive acceleration in solar and wind investment.")
            ),
            div(class = "event-card",
              div(class = "event-year-card", "2014"),
              div(class = "event-title-card", "China Energy Strategy"),
```

```
                    div(class = "event-desc-card", "National Energy Strategy Action Plan: Cap on coal use, solar/wind targets, major policy shift.")
                ),
                div(class = "event-card",
                    div(class = "event-year-card", "2015"),
                    div(class = "event-title-card", "Paris Agreement"),
                    div(class = "event-desc-card", "Global climate accord uniting nations in commitment to limit warming to well below 2°C.")
                ),
                div(class = "event-card",
                    div(class = "event-year-card", "2021"),
                    div(class = "event-title-card", "China Carbon Pledge"),
                    div(class = "event-desc-card", "Carbon neutrality pledge targeting net zero by 2060; strong acceleration in solar/wind since.")
                ),
                div(class = "event-card",
                    div(class = "event-year-card", "2022"),
                    div(class = "event-title-card", "US IRA"),
                    div(class = "event-desc-card", "Inflation Reduction Act: $370B in climate and energy investments — the largest climate legislation in US history.")
                )
              )
            )
          )
        )
      ),

      # Impact Analysis Tab
      tabItem(
        tabName = "impact",
```

```
fluidRow(
  column(12,
    div(class = "section-header",
      h3("Renewable Energy and CO₂ Emissions"),
      p("Examining the relationship between renewable energy adoption and carbon intensity of electricity generation"),
      div(class = "note-box",
        icon("info-circle"),
        "Countries with higher renewable shares typically show lower CO₂ intensity.
        Bubble size represents per capita energy consumption. The trend line demonstrates the inverse relationship between renewable adoption and emissions."
      )
    )
  )
),
fluidRow(
  column(8,
    div(class = "viz-container",
      plotlyOutput("scatterPlot", height = "450px")
    )
  ),
  column(4,
    div(class = "viz-container",
      h4("Regional Averages"),
      plotlyOutput("regionalPlot", height = "450px")
    )
  )
),
fluidRow(
```

```
        column(12,
            div(class = "insights-container",
              h4("Key Insights from the Analysis"),
              div(class = "insight-box",
                p("• ", strong("Accelerating Global Transition:"), " The global average renewable energy share has increased from 17.3% in 2000 to over 30% in recent years, with acceleration particularly notable after the Paris Agreement (2015)."),
                p("• ", strong("Policy Impact:"), " Major policy interventions show clear correlations with renewable energy uptake. Germany's EEG (2000) and China's strategic shift (2014) demonstrate how targeted policies drive transformation."),
                p("• ", strong("Emissions Reduction:"), " Countries achieving >50% renewable electricity show 60-80% lower CO₂ intensity compared to fossil-dependent nations, validating renewable energy as a climate solution."),
                p("• ", strong("Regional Variations:"), " Europe and South America lead with 40%+ renewable shares, while Asia shows the fastest growth trajectory despite starting from a lower base.")
              )
            )
        )
      )
    )
  )
)

# Server Logic

server <- function(input, output, session) {

  # Color palettes
  gg_colors <- scales::hue_pal()(6)
  watercolor_palette <- paste0(gg_colors, "80")
```

```
country_colors <- c(
  "China" = "#FF998080",
  "United States" = "#99CCFF80",
  "Germany" = "#99FF9980",
  "India" = "#FFCC9980",
  "Brazil" = "#FF99CC80",
  "Norway" = "#CCCCFF80",
  "Japan" = "#FFFF9980"
)

region_colors_palette <- scales::hue_pal()(6)

# Reactive data
yearData <- reactive({
  filter(energy, year == input$year)
})

# Global average
output$globalAvgText <- renderText({
  avg <- yearData() %>%
    summarise(avg = mean(renewables_share_elec, na.rm = TRUE)) %>%
    pull(avg)
  paste0(round(avg, 1), "%")
})

# Statistics
output$statCountries <- renderText({
  nrow(yearData())
```

```
})

output$statLeaders <- renderText({
  sum(yearData()$renewables_share_elec > 50, na.rm = TRUE)
})

output$statGrowth <- renderText({
  if (input$year > 2000) {
    prev_year <- filter(energy, year == input$year - 1)
    curr_year <- yearData()

    prev_avg <- mean(prev_year$renewables_share_elec, na.rm = TRUE)
    curr_avg <- mean(curr_year$renewables_share_elec, na.rm = TRUE)
    growth <- curr_avg - prev_avg

    paste0(ifelse(growth > 0, "+", ""), round(growth, 1), "%")
  } else {
    "N/A"
  }
})

# World map
output$worldMap <- renderPlotly({
  df <- yearData()

  colors <- list(
    c(0, 0.2, 0.4, 0.6, 0.8, 1),
    c("#FFE5CC", "#FFD4A3", "#B8E6B8", "#8FD68F", "#66C266", "#2E8B2E")
  )
```

```
plot_geo(data = df, locationmode = 'ISO-3') %>%
  add_trace(
    type = 'choropleth',
    locations = ~iso_code,
    z = ~renewables_share_elec,
    text = ~paste0(country, "\n", "Renewable: ", round(renewables_share_elec, 1), "%"),
    hoverinfo = "text",
    colorscale = colors,
    reversescale = FALSE,
    marker = list(line = list(width = 0.5, color = 'rgba(255,255,255,0.5)')),
    colorbar = list(
      title = list(text = "% Renewable", font = list(family = "Gill Sans, sans-serif", size = 12)),
      thickness = 12,
      len = 0.6,
      x = 0.98,
      tickfont = list(family = "Gill Sans, sans-serif", size = 10)
    )
  ) %>%
  layout(
    geo = list(
      projection = list(type = "natural earth"),
      showframe = FALSE,
      showcoastlines = TRUE,
      coastlinecolor = "rgba(100,100,100,0.3)",
      bgcolor = "rgba(245,245,245,0.3)",
      showocean = TRUE,
      oceancolor = "rgba(173,216,230,0.3)",
      showlakes = TRUE,
```

```
      lakecolor = "rgba(173,216,230,0.3)"
    ),
    margin = list(l = 0, r = 0, t = 0, b = 0),
    font = list(family = "Gill Sans, sans-serif")
  ) %>%
  config(displayModeBar = FALSE)
})

# Time series plot
output$timeSeriesPlot <- renderPlotly({
  countries <- c("China", "United States", "Germany", "India", "Brazil", "Norway")
  df <- filter(energy, country %in% countries)

  events <- data.frame(
    year = c(2011, 2014, 2015, 2021, 2022),
    label = c("Fukushima", "China\nStrategy", "Paris\nAgreement", "China\nNet Zero", "US\nIRA"),
    y_pos = c(85, 90, 85, 90, 85),
    color = "#66666640"
  )

  p <- plot_ly()

  # Event lines
  for(i in seq_len(nrow(events))) {
    p <- p %>%
      add_trace(
        x = c(events$year[i], events$year[i]),
        y = c(0, 100),
        type = 'scatter',
```

```
      mode = 'lines',
      line = list(color = events$color[i], width = 1, dash = 'dash'),
      showlegend = FALSE,
      hoverinfo = 'skip'
    ) %>%
    add_annotations(
      x = events$year[i],
      y = events$y_pos[i],
      text = events$label[i],
      showarrow = FALSE,
      font = list(size = 10, color = "#666666", family = "Gill Sans, sans-serif"),
      bgcolor = "rgba(255,255,255,0)",
      bordercolor = "rgba(0,0,0,0)",
      yanchor = "bottom",
      yshift = 5
    )
}

# Country lines
last_year_data <- df %>%
  group_by(country) %>%
  filter(year == max(year)) %>%
  ungroup()

for(i in seq_along(countries)) {
  df_country <- filter(df, country == countries[i])
  p <- p %>%
    add_trace(
      data = df_country,
```

```
      x = ~year,
      y = ~renewables_share_elec,
      name = countries[i],
      type = 'scatter',
      mode = 'lines',
      line = list(width = 3, color = country_colors[countries[i]]),
      opacity = 0.9,
      hovertemplate = paste0(countries[i], "<br>Year: %{x}<br>Renewable:
%{y:.1f}%<br><extra></extra>"),
      showlegend = FALSE
    )
  }

  # End labels
  p <- p %>%
    add_trace(
      data = last_year_data,
      x = ~year,
      y = ~renewables_share_elec,
      text = ~country,
      type = 'scatter',
      mode = 'text',
      textposition = 'middle right',
      textfont = list(
        size = 11,
        color = sapply(last_year_data$country, function(c) country_colors[c]),
        family = "Gill Sans, sans-serif"
      ),
      showlegend = FALSE,
```

```
      hoverinfo = 'skip'
    )

  p %>%
    layout(
      xaxis = list(
        title = "Year",
        gridcolor = "rgba(240,240,240,0.5)",
        font = list(family = "Gill Sans, sans-serif", size = 12),
        range = c(2000, max(df$year) + 2)
      ),
      yaxis = list(
        title = "Renewable Energy Share (%)",
        range = c(0, 100),
        gridcolor = "rgba(240,240,240,0.5)",
        font = list(family = "Gill Sans, sans-serif", size = 12)
      ),
      margin = list(l = 60, r = 100, t = 20, b = 40),
      plot_bgcolor = "rgba(255,255,255,0.8)",
      paper_bgcolor = "white",
      font = list(family = "Gill Sans, sans-serif")
    ) %>%
    config(displayModeBar = FALSE)
})

# Progress plot
output$progressPlot <- renderPlotly({
  countries <- c("China", "United States", "Germany", "India", "Brazil", "Japan")
```

```
  df_comparison <- energy %>%
    filter(country %in% countries, year %in% c(2000, input$year)) %>%
    select(country, year, renewables_share_elec) %>%
    pivot_wider(names_from = year, values_from = renewables_share_elec, names_prefix = "year_")
%>%
    mutate(
      change = get(paste0("year_", input$year)) - year_2000,
      country = factor(country, levels = country[order(change)])
    )

  bar_colors <- sapply(as.character(df_comparison$country), function(c) country_colors[c])

  plot_ly(df_comparison) %>%
    add_trace(
      x = ~change,
      y = ~country,
      type = 'bar',
      orientation = 'h',
      marker = list(
        color = bar_colors,
        line = list(color = 'rgba(255,255,255,0.5)', width = 1)
      ),
      opacity = 0.9,
      text = ~paste0(round(change, 1), "%"),
      textposition = "outside",
      textfont = list(size = 10, family = "Gill Sans, sans-serif"),
      hovertemplate = paste0("%{y}<br>Change: %{x:+.1f}%<br><extra></extra>")
    ) %>%
    layout(
```

```
      xaxis = list(
        title = "Change (%)",
        gridcolor = "rgba(240,240,240,0.5)",
        font = list(family = "Gill Sans, sans-serif", size = 10)
      ),
      yaxis = list(
        title = "",
        font = list(family = "Gill Sans, sans-serif", size = 10)
      ),
      margin = list(l = 80, r = 40, t = 10, b = 30),
      plot_bgcolor = "rgba(255,255,255,0.8)",
      paper_bgcolor = "white",
      font = list(family = "Gill Sans, sans-serif")
    ) %>%
    config(displayModeBar = FALSE)
})

# Scatter plot
output$scatterPlot <- renderPlotly({
  df <- yearData() %>%
    filter(!is.na(co2_intensity), co2_intensity > 0) %>%
    filter(co2_intensity < quantile(co2_intensity, 0.95, na.rm = TRUE))

  fit <- lm(log10(co2_intensity) ~ renewables_share_elec, data = df)
  x_range <- seq(0, 100, by = 1)
  y_pred <- 10^predict(fit, newdata = data.frame(renewables_share_elec = x_range))

  # Define watercolor palette for regions
  region_colors <- c(
```

```
  "Africa" = "rgba(255, 179, 186, 0.7)",      # Soft pink
  "Americas" = "rgba(186, 225, 255, 0.7)",    # Soft blue
  "Asia" = "rgba(255, 223, 186, 0.7)",        # Soft peach
  "Europe" = "rgba(186, 255, 201, 0.7)",      # Soft mint green
  "Oceania" = "rgba(221, 186, 255, 0.7)"      # Soft lavender
)

p <- plot_ly() %>%
  add_trace(
    x = x_range,
    y = y_pred,
    type = 'scatter',
    mode = 'lines',
    line = list(color = 'rgba(100,100,100,0.5)', width = 2, dash = 'dash'),
    showlegend = FALSE,
    hoverinfo = 'skip'
  )

# Add each region as a separate trace to maintain consistent colors
for(region_name in names(region_colors)) {
  df_region <- df %>% filter(region == region_name)

  if(nrow(df_region) > 0) {
    p <- p %>%
      add_trace(
        data = df_region,
        x = ~renewables_share_elec,
        y = ~co2_intensity,
        type = 'scatter',
```

```
      mode = 'markers',
      name = region_name,
      marker = list(
        size = ~sqrt(energy_per_capita),
        sizemode = 'area',
        sizeref = 2,
        color = region_colors[region_name],
        line = list(color = 'rgba(255,255,255,0.8)', width = 1)
      ),
      text = ~paste0(country, "\n",
                     "Renewable: ", round(renewables_share_elec, 1), "%\n",
                     "CO₂: ", round(co2_intensity, 0), " t/TWh"),
      hoverinfo = "text"
    )
  }
}

p %>%
  layout(
    xaxis = list(
      title = "Renewable Energy Share (%)",
      range = c(-5, 105),
      gridcolor = "rgba(240,240,240,0.5)",
      font = list(family = "Gill Sans, sans-serif", size = 12)
    ),
    yaxis = list(
      title = "CO₂ Intensity (tonnes per TWh) - Log Scale",
      type = "log",
      gridcolor = "rgba(240,240,240,0.5)",
```

```
      font = list(family = "Gill Sans, sans-serif", size = 12)
    ),
    legend = list(
      title = list(text = "Region", font = list(family = "Gill Sans, sans-serif", size = 11)),
      font = list(family = "Gill Sans, sans-serif", size = 10),
      bgcolor = "rgba(255,255,255,0.8)",
      bordercolor = "rgba(200,200,200,0.5)",
      borderwidth = 1
    ),
    margin = list(l = 80, r = 20, t = 20, b = 60),
    plot_bgcolor = "rgba(255,255,255,0.8)",
    paper_bgcolor = "white",
    font = list(family = "Gill Sans, sans-serif")
  ) %>%
  config(displayModeBar = FALSE)
})

# Regional plot
output$regionalPlot <- renderPlotly({
  df <- yearData() %>%
    group_by(region) %>%
    summarise(
      renewable_avg = mean(renewables_share_elec, na.rm = TRUE),
      .groups = 'drop'
    ) %>%
    arrange(desc(renewable_avg))

  # Use same watercolor palette as scatter plot
  region_colors <- c(
```

```
  "Africa" = "rgba(255, 179, 186, 0.7)",      # Soft pink
  "Americas" = "rgba(186, 225, 255, 0.7)",    # Soft blue
  "Asia" = "rgba(255, 223, 186, 0.7)",        # Soft peach
  "Europe" = "rgba(186, 255, 201, 0.7)",      # Soft mint green
  "Oceania" = "rgba(221, 186, 255, 0.7)"      # Soft lavender
)

plot_ly(df,
        x = ~renewable_avg,
        y = ~reorder(region, renewable_avg),
        type = 'bar',
        orientation = 'h',
        marker = list(
          color = sapply(df$region, function(r) region_colors[r]),
          line = list(color = 'rgba(255,255,255,0.5)', width = 1)
        ),
        opacity = 0.9,
        text = ~paste0(round(renewable_avg, 1), "%"),
        textposition = "outside",
        textfont = list(size = 10, family = "Gill Sans, sans-serif"),
        hovertemplate = paste0("%{y}<br>Average: %{x:.1f}%<br><extra></extra>")
) %>%
  layout(
    xaxis = list(
      title = "Average Renewable %",
      range = c(0, 60),
      gridcolor = "rgba(240,240,240,0.5)",
      font = list(family = "Gill Sans, sans-serif", size = 11)
    ),
```

```
      yaxis = list(
        title = "",
        font = list(family = "Gill Sans, sans-serif", size = 11)
      ),
      margin = list(l = 80, r = 40, t = 20, b = 40),
      plot_bgcolor = "rgba(255,255,255,0.8)",
      paper_bgcolor = "white",
      font = list(family = "Gill Sans, sans-serif")
    ) %>%
    config(displayModeBar = FALSE)
  })
}

# Run Application
shinyApp(ui, server)
```

style.css

```
/* Global Energy Transition Dashboard */

/* Base Settings */
* {
  box-sizing: border-box;
}

html, body {
  height: 1080px;
  width: 1920px;
  overflow: hidden;
  margin: 0;
```

```
  padding: 0;
  font-family: 'Gill Sans', sans-serif;
  font-size: 14px;
  color: #333333;
  background-color: #ffffff;
}

/* Layout Structure */
.wrapper {
  height: 1080px !important;
  width: 1920px !important;
  overflow: hidden !important;
}

.content-wrapper, .right-side {
  height: 1080px !important;
  overflow: hidden !important;
  margin-left: 250px !important;
  background-color: rgba(250, 250, 250, 0.5);
}

/* Header */
.main-header {
  position: fixed;
  width: 100%;
  z-index: 1000;
}

.main-header .logo {
```

```
  background-color: rgba(255, 255, 255, 0.95) !important;
  color: #333333 !important;
  font-weight: 400;
  font-size: 19px;
  letter-spacing: 0.5px;
  border-bottom: 1px solid rgba(224, 224, 224, 0.5);
  font-family: 'Gill Sans', sans-serif;
}

.main-header .navbar {
  background-color: rgba(255, 255, 255, 0.95) !important;
  margin-left: 250px !important;
  border-bottom: 1px solid rgba(224, 224, 224, 0.5);
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);
}

/* Sidebar */
.main-sidebar {
  height: 1080px !important;
  position: fixed !important;
  overflow-y: auto !important;
  overflow-x: hidden !important;
  background: linear-gradient(to bottom, rgba(248, 248, 248, 0.95), rgba(245, 245, 245, 0.9)) !important;
  padding-top: 50px;
  border-right: 1px solid rgba(224, 224, 224, 0.5);
}

.sidebar-menu > li > a {
  color: #333333 !important;
```

```
  padding: 12px 20px !important;
  font-size: 14px;
  font-weight: 400;
  transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
  font-family: 'Gill Sans', sans-serif;
}

.sidebar-menu > li.active > a,
.sidebar-menu > li > a:hover {
  background-color: rgba(255, 255, 255, 0.9) !important;
  border-left: 3px solid rgba(141, 211, 199, 0.9) !important;
  padding-left: 17px !important;
  transform: translateX(2px);
}

.sidebar-menu > li > a i {
  margin-right: 10px;
  font-size: 14px;
  opacity: 0.7;
  transition: opacity 0.3s ease;
}

.sidebar-menu > li:hover > a i {
  opacity: 1;
}

/* Dashboard Description */
.dashboard-description {
  padding: 15px 20px;
```

```
  margin: 10px 15px;
  background-color: rgba(255, 255, 255, 0.7);
  border-radius: 8px;
  font-size: 13px;
  line-height: 1.6;
  color: #555555;
  font-family: 'Gill Sans', sans-serif;
  animation: fadeIn 0.8s ease-out;
}

/* Content Area */
.content {
  padding: 70px 20px 20px 20px !important;
  height: calc(1080px - 50px) !important;
  overflow: hidden !important;
}

.tab-content {
  height: calc(100% - 20px) !important;
  overflow: hidden !important;
}

.tab-pane {
  height: 100% !important;
  overflow: hidden !important;
}

/* Year Container with Slider */
.year-container {
```

```
  padding: 20px;
  margin: 15px;
  background-color: rgba(255, 255, 255, 0.85);
  border: 1px solid rgba(224, 224, 224, 0.5);
  border-radius: 8px;
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.08);
  animation: slideIn 0.6s ease-out;
  position: relative;
}

.year-container h5 {
  margin: 0 0 15px 0;
  font-size: 12px;
  font-weight: 600;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  color: #666666;
  font-family: 'Gill Sans', sans-serif;
}

/* Play button positioning */
.btn-play-slider {
  position: absolute;
  bottom: 20px;
  left: 20px;
  width: 35px;
  height: 35px;
  border-radius: 50%;
  background-color: rgba(141, 211, 199, 0.8) !important;
```

```
  border: 2px solid rgba(255, 255, 255, 0.9) !important;
  color: white !important;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.15);
  padding: 0 !important;
  font-size: 14px;
}

.btn-play-slider:hover {
  background-color: rgba(141, 211, 199, 1) !important;
  transform: scale(1.1);
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
}

.btn-play-slider:active {
  transform: scale(0.95);
}

.btn-play-slider:focus {
  outline: none !important;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2) !important;
}

/* Metric Container with light green for global average */
.metric-container {
```

```
  padding: 20px;
  margin: 15px;
  background-color: rgba(51, 51, 51, 0.9);
  color: #ffffff;
  border-radius: 8px;
  box-shadow: 0 3px 8px rgba(0, 0, 0, 0.12);
  transition: all 0.3s ease;
  animation: pulseIn 0.8s ease-out;
}

.metric-container.global-average {
  background: linear-gradient(135deg, rgba(152, 251, 152, 0.25), rgba(144, 238, 144, 0.35));
  color: #2d5a2d;
  border: 1px solid rgba(152, 251, 152, 0.4);
}

.metric-container h5 {
  margin: 0 0 10px 0;
  font-size: 12px;
  font-weight: 400;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  opacity: 0.8;
  font-family: 'Gill Sans', sans-serif;
}

.global-average h5 {
  color: #2d5a2d;
}
```

```
.metric-value {
  font-size: 42px;
  font-weight: 300;
  line-height: 1;
  margin-bottom: 5px;
  font-family: 'Gill Sans', sans-serif;
  animation: countUp 1s ease-out;
}

.metric-label {
  font-size: 12px;
  opacity: 0.8;
  font-family: 'Gill Sans', sans-serif;
}

/* Sidebar Footer */
.sidebar-footer {
  position: absolute;
  bottom: 120px;
  left: 0;
  right: 0;
  padding: 15px 20px;
  border-top: 1px solid rgba(224, 224, 224, 0.3);
  background-color: rgba(255, 255, 255, 0.5);
}

.sidebar-footer p {
  margin: 2px 0;
```

```
  font-size: 11px;
  color: #888888;
  font-family: 'Gill Sans', sans-serif;
}

/* Section Headers - Compact */
.section-header {
  margin-bottom: 15px;
  padding-bottom: 10px;
  border-bottom: 1px solid rgba(224, 224, 224, 0.5);
  animation: fadeInDown 0.6s ease-out;
}

.section-header h3 {
  margin: 0 0 5px 0;
  font-size: 24px;
  font-weight: 400;
  color: #333333;
  font-family: 'Gill Sans', sans-serif;
}

.section-subtitle {
  font-size: 15px;
  color: #444444;
  font-weight: 500;
  line-height: 1.4;
  margin-bottom: 6px;
  font-family: 'Gill Sans', sans-serif;
}
```

```
.section-header p {
  margin: 0 0 3px 0;
  font-size: 13px;
  color: #666666;
  line-height: 1.4;
  font-family: 'Gill Sans', sans-serif;
}

.section-description {
  font-size: 12px !important;
  color: #555555 !important;
  font-style: italic;
  margin-top: 6px !important;
  background-color: rgba(245, 245, 245, 0.5);
  padding: 8px 12px;
  border-radius: 6px;
  border-left: 3px solid rgba(141, 211, 199, 0.3);
}

/* Note Box */
.note-box {
  margin-top: 10px;
  padding: 14px 20px;
  background: linear-gradient(135deg, rgba(245, 245, 245, 0.6), rgba(240, 248, 255, 0.4));
  border-left: 3px solid rgba(141, 211, 199, 0.7);
  font-size: 13px;
  color: #555555;
  line-height: 1.7;
```

```
  border-radius: 6px;
  font-family: 'Gill Sans', sans-serif;
  animation: slideInLeft 0.6s ease-out;
}

.note-box i {
  margin-right: 8px;
  opacity: 0.8;
  color: rgba(141, 211, 199, 0.9);
}

/* Visualization Containers - Compact */
.main-viz-container,
.viz-container {
  background-color: rgba(255, 255, 255, 0.92);
  border: 1px solid rgba(224, 224, 224, 0.3);
  border-radius: 10px;
  padding: 15px;
  margin-bottom: 10px;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
  position: relative;
  overflow: hidden;
  animation: zoomIn 0.7s ease-out;
}

.viz-container.compact {
  padding: 12px;
}
```

```
.viz-container h4 {
  margin: 0 0 10px 0;
  font-size: 14px;
  font-weight: 600;
  color: #333333;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  font-family: 'Gill Sans', sans-serif;
}

.viz-container h4 {
  margin: 0 0 15px 0;
  font-size: 16px;
  font-weight: 600;
  color: #333333;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  font-family: 'Gill Sans', sans-serif;
}

/* Stat Boxes */
.stat-box {
  background: linear-gradient(135deg, rgba(255, 255, 255, 0.9), rgba(250, 250, 250, 0.8));
  border: 1px solid rgba(224, 224, 224, 0.3);
  border-radius: 10px;
  padding: 30px;
  text-align: center;
  margin-bottom: 20px;
  box-shadow: 0 3px 10px rgba(0, 0, 0, 0.08);
```

```
  transition: all 0.4s cubic-bezier(0.4, 0, 0.2, 1);
  animation: bounceIn 0.8s ease-out;
}

.stat-box:hover {
  background: linear-gradient(135deg, rgba(141, 211, 199, 0.15), rgba(141, 211, 199, 0.08));
  border-color: rgba(141, 211, 199, 0.4);
  transform: translateY(-5px);
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.12);
}

.stat-value {
  font-size: 36px;
  font-weight: 300;
  color: #333333;
  margin-bottom: 5px;
  font-family: 'Gill Sans', sans-serif;
  transition: color 0.3s ease;
}

.stat-box:hover .stat-value {
  color: #2d5a2d;
}

.stat-label {
  font-size: 13px;
  color: #666666;
  text-transform: uppercase;
  letter-spacing: 0.5px;
```

```
  font-family: 'Gill Sans', sans-serif;
}

/* Events Timeline Strip */
.events-strip {
  background: linear-gradient(to right, rgba(245, 245, 245, 0.8), rgba(250, 250, 250, 0.8));
  border-radius: 10px;
  padding: 20px;
  margin: 15px 0;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.06);
  animation: expandIn 0.8s ease-out;
}

.events-strip h4 {
  margin: 0 0 20px 0;
  font-size: 16px;
  font-weight: 600;
  color: #333333;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  font-family: 'Gill Sans', sans-serif;
  text-align: center;
}

.events-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 15px;
  padding: 0 10px;
```

```
}

.event-card {
  background: rgba(255, 255, 255, 0.9);
  border-left: 4px solid rgba(141, 211, 199, 0.8);
  border-radius: 6px;
  padding: 15px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);
  transition: all 0.3s ease;
  animation: fadeInUp 0.6s ease-out;
}

.event-card:hover {
  transform: translateY(-2px);
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  background: rgba(141, 211, 199, 0.1);
}

.event-year-card {
  font-size: 14px;
  font-weight: 700;
  color: #2d5a2d;
  margin-bottom: 4px;
  font-family: 'Gill Sans', sans-serif;
}

.event-title-card {
  font-size: 13px;
  font-weight: 600;
```

```
  color: #333333;
  margin-bottom: 6px;
  font-family: 'Gill Sans', sans-serif;
}

.event-desc-card {
  font-size: 12px;
  color: #555555;
  line-height: 1.5;
  font-family: 'Gill Sans', sans-serif;
}

/* Insights Container - Compact */
.insights-container {
  background: linear-gradient(135deg, rgba(255, 255, 255, 0.95), rgba(245, 250, 245, 0.9));
  border: 1px solid rgba(224, 224, 224, 0.3);
  border-radius: 10px;
  padding: 15px 20px !important;
  margin-top: 10px;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
  animation: fadeInUp 0.8s ease-out;
}

.insights-container h4 {
  margin: 0 0 10px 0;
  font-size: 15px;
  font-weight: 600;
  color: #2d5a2d;
  text-transform: uppercase;
```

```
  letter-spacing: 0.5px;
  font-family: 'Gill Sans', sans-serif;
  text-align: left;
}

.insight-box {
  background-color: rgba(255, 255, 255, 0.7);
  padding: 10px 15px;
  border-radius: 8px;
  border: 1px solid rgba(141, 211, 199, 0.2);
}

.insight-box p {
  margin: 0 0 8px 0;
  font-size: 13px;
  line-height: 1.5;
  color: #444444;
  font-family: 'Gill Sans', sans-serif;
}

.insight-box p:last-child {
  margin-bottom: 0;
}

.insight-box strong {
  color: #2d5a2d;
  font-weight: 600;
}
```

```
.conclusion {
  margin-top: 20px !important;
  padding-top: 20px;
  border-top: 1px solid rgba(141, 211, 199, 0.3);
  font-style: italic;
}

/* Slider Styling - Enhanced with ruler marks */
.irs--shiny {
  font-family: 'Gill Sans', sans-serif;
  height: 75px;
  margin-bottom: 20px;
}

.irs {
  margin-top: 25px;
}

.irs-bar {
  background: linear-gradient(to right, rgba(255, 200, 100, 0.6), rgba(141, 211, 199, 0.8)) !important;
  border: none !important;
  height: 8px !important;
  top: 30px !important;
  border-radius: 4px;
}

.irs-bar-edge {
  background: rgba(141, 211, 199, 0.8) !important;
  border: none !important;
```

```
  border-radius: 4px 0 0 4px;
}

.irs-handle {
  background: rgba(141, 211, 199, 0.95) !important;
  border: 3px solid rgba(255, 255, 255, 0.95) !important;
  box-shadow: 0 3px 8px rgba(0, 0, 0, 0.25) !important;
  width: 26px !important;
  height: 26px !important;
  top: 21px !important;
  cursor: grab;
  transition: all 0.2s ease;
}

.irs-handle:hover {
  background: rgba(141, 211, 199, 1) !important;
  transform: scale(1.1);
}

.irs-handle:active {
  cursor: grabbing;
  transform: scale(0.95);
}

.irs-single {
  background: rgba(141, 211, 199, 0.9) !important;
  color: #ffffff !important;
  font-size: 13px !important;
  padding: 4px 10px !important;
```

```
  font-family: 'Gill Sans', sans-serif;
  border-radius: 4px;
  font-weight: 600;
  top: 0px !important;
}

.irs-grid {
  height: 27px !important;
}

.irs-grid-pol {
  background: rgba(150, 150, 150, 0.4) !important;
  height: 10px !important;
  width: 1px !important;
  top: 37px !important;
}

.irs-grid-pol.small {
  height: 5px !important;
  top: 40px !important;
}

.irs-grid-text {
  color: #555555 !important;
  font-size: 11px !important;
  font-family: 'Gill Sans', sans-serif;
  bottom: 15px !important;
  font-weight: 500;
}
```

```
/* Box Styling */
.box {
  border: none !important;
  box-shadow: none !important;
  background-color: transparent !important;
  margin-bottom: 0 !important;
}

.box-header {
  display: none !important;
}

.box-body {
  padding: 0 !important;
}

/* Remove Plotly Modebar */
.plotly .modebar {
  display: none !important;
}

/* Scrollbar */
::-webkit-scrollbar {
  width: 10px;
  height: 10px;
}

::-webkit-scrollbar-track {
```

```
  background: rgba(245, 245, 245, 0.5);
  border-radius: 5px;
}

::-webkit-scrollbar-thumb {
  background: linear-gradient(to bottom, rgba(141, 211, 199, 0.6), rgba(141, 211, 199, 0.4));
  border-radius: 5px;
  border: 1px solid rgba(255, 255, 255, 0.3);
}

::-webkit-scrollbar-thumb:hover {
  background: linear-gradient(to bottom, rgba(141, 211, 199, 0.8), rgba(141, 211, 199, 0.6));
}

/* Watercolor effect overlays */
.main-viz-container::before,
.viz-container::before {
  content: '';
  position: absolute;
  top: -50%;
  left: -50%;
  right: -50%;
  bottom: -50%;
  background-image:
    radial-gradient(circle at 20% 80%, rgba(141, 211, 199, 0.08) 0%, transparent 50%),
    radial-gradient(circle at 80% 20%, rgba(255, 220, 180, 0.06) 0%, transparent 50%),
    radial-gradient(circle at 40% 40%, rgba(190, 186, 218, 0.06) 0%, transparent 50%);
  pointer-events: none;
  z-index: 1;
```

```
  transform: rotate(-5deg);
}

/* Content should be above watercolor overlay */
.main-viz-container > *,
.viz-container > * {
  position: relative;
  z-index: 2;
}

/* Animations */
@keyframes fadeIn {
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}

@keyframes fadeInDown {
  from {
    opacity: 0;
    transform: translateY(-20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
```

```
}

@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

@keyframes slideIn {
  from {
    opacity: 0;
    transform: translateX(-20px);
  }
  to {
    opacity: 1;
    transform: translateX(0);
  }
}

@keyframes slideInLeft {
  from {
    opacity: 0;
    transform: translateX(-30px);
  }
```

```
  to {
    opacity: 1;
    transform: translateX(0);
  }
}

@keyframes zoomIn {
  from {
    opacity: 0;
    transform: scale(0.95);
  }
  to {
    opacity: 1;
    transform: scale(1);
  }
}

@keyframes bounceIn {
  0% {
    opacity: 0;
    transform: scale(0.3);
  }
  50% {
    opacity: 1;
    transform: scale(1.05);
  }
  70% {
    transform: scale(0.95);
  }
```

```
  100% {
    transform: scale(1);
  }
}

@keyframes expandIn {
  from {
    opacity: 0;
    transform: scaleX(0.8);
  }
  to {
    opacity: 1;
    transform: scaleX(1);
  }
}

@keyframes pulseIn {
  0% {
    opacity: 0.6;
    transform: scale(0.98);
  }
  50% {
    opacity: 0.8;
    transform: scale(1.02);
  }
  100% {
    opacity: 1;
    transform: scale(1);
  }
```

```
}

@keyframes countUp {
  from {
    opacity: 0;
    transform: translateY(10px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

/* Apply staggered animations to elements */
.stat-box:nth-child(1) { animation-delay: 0.1s; }
.stat-box:nth-child(2) { animation-delay: 0.2s; }
.stat-box:nth-child(3) { animation-delay: 0.3s; }

.event-compact:nth-child(1) { animation: fadeInUp 0.6s ease-out 0.1s both; }
.event-compact:nth-child(2) { animation: fadeInUp 0.6s ease-out 0.2s both; }
.event-compact:nth-child(3) { animation: fadeInUp 0.6s ease-out 0.3s both; }
.event-compact:nth-child(4) { animation: fadeInUp 0.6s ease-out 0.4s both; }
.event-compact:nth-child(5) { animation: fadeInUp 0.6s ease-out 0.5s both; }
.event-compact:nth-child(6) { animation: fadeInUp 0.6s ease-out 0.6s both; }
.event-compact:nth-child(7) { animation: fadeInUp 0.6s ease-out 0.7s both; }

/* Responsive adjustments for 1920x1080 */
.row {
  margin-left: -10px;
```

```
  margin-right: -10px;
}

.col-sm-1, .col-sm-2, .col-sm-3, .col-sm-4, .col-sm-5, .col-sm-6,
.col-sm-7, .col-sm-8, .col-sm-9, .col-sm-10, .col-sm-11, .col-sm-12 {
  padding-left: 10px;
  padding-right: 10px;
}

/* Clean table styling */
.table {
  font-size: 13px;
  border: 1px solid rgba(224, 224, 224, 0.3);
  font-family: 'Gill Sans', sans-serif;
}

.table-striped > tbody > tr:nth-of-type(odd) {
  background-color: rgba(250, 250, 250, 0.5);
}

.table-hover > tbody > tr:hover {
  background-color: rgba(141, 211, 199, 0.1);
}

/* Ensure plotly plots fill their containers */
.plotly {
  width: 100% !important;
  height: 100% !important;
}
```

```
/* Additional polish */
h3, h4, h5 {
  text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.05);
}

/* Make sure all text uses Gill Sans */
* {
  font-family: 'Gill Sans', sans-serif !important;
}

/* Final touches for publication quality */
.main-viz-container,
.viz-container {
  backdrop-filter: blur(2px);
}

/* Smooth transitions for all interactive elements */
a, button, input, select, .stat-box, .event-compact {
  transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
}

/* Print-ready adjustments */
@media print {
  .main-sidebar, .main-header {
    display: none !important;
  }

  .content-wrapper {
```

```
    margin-left: 0 !important;
  }

  * {
    animation: none !important;
    transition: none !important;
  }
}
```