Tourism Recovery: A Multi-Method Analysis of Visitors, Spending, and Media Narratives

Hung Tran, **Joseph** Martinez and **Tanmoy** Biswas

University of Texas at Dallas

EPPS 6356: Data Visualization

Instructor: Dr. Karl Ho

December 5, 2025

Abstract

This project examines post-pandemic tourism recovery in the United States through three complementary lenses: administrative tax data from California, regional spending patterns in New York, and national media narratives. Using interactive Shiny dashboards, we visualize Transient Occupancy Tax (TOT) revenues across California counties, compare Adirondacks regional spending against statewide New York benchmarks, and analyze sentiment and topic patterns in tourism-related news coverage from 2019 to 2025.

Our findings suggest that tourism spending and visitor activity largely move together once data are properly cleaned, indexed, and interpreted, challenging the popular narrative of "crowds without cash." The Adirondacks region shows particularly strong recovery, reaching 161.8% of its 2019 spending level by 2024 and outpacing New York State overall. Media sentiment follows a similar trajectory, turning increasingly positive after 2021 as destinations reopen. Taken together, these visualizations demonstrate how triangulating fiscal records, regional indices, and narrative framing can clarify whether perceived tourism paradoxes reflect genuine economic divergence or artifacts of data structure, aggregation, and media storytelling.

Keywords: tourism recovery, data visualization, sentiment analysis, topic modeling, transient occupancy tax

Introduction

Tourism is one of the clearest indicators of economic and social recovery after major shocks. The COVID-19 pandemic disrupted travel globally, created sharp declines in mobility, and altered how people chose destinations. As the United States emerged from the pandemic, a recurring narrative appeared across news outlets: visitor volumes seemed to rebound quickly, yet spending did not keep pace. Headlines suggested that parks, cities, and recreational destinations were crowded, but hotel taxes and tourism revenues lagged behind. This idea—often described as a "tourism paradox"—became a widely circulated storyline.

Our project revisits this claim by examining whether visitors and spending actually diverged in meaningful ways, or whether the perceived mismatch is better explained by changes in where people travel, how data are reported, or how media frame recovery. To do this, our team developed three interactive dashboards using independent datasets from California, New York, and national news coverage. Each dashboard contributes a different angle: administrative tax receipts, regional spending indices, and media sentiment and topic patterns. Together, these three perspectives allow us to evaluate whether the underlying data support the paradox narrative or whether spending and visitor activity largely move together.

This approach reflects the broader shift toward data visualization as a core method for policy-relevant analysis. Rather than treating tourism as a single headline metric, we focus on the structure of recovery within states, across rural and urban regions, and within the national conversation. By integrating quantitative patterns with narrative framing, our project shows how visual tools can clarify whether public perceptions match underlying economic behavior.

Methods

Overview

Our methodological approach follows the framework laid out in our project proposal. We examine tourism recovery through three distinct but complementary datasets. Each team member designed a stand-alone Shiny dashboard, and our final analysis synthesizes patterns across these dashboards. Although each visualization focuses on a different geographic unit, all three operationalize the same core idea: evaluate how spending behaves relative to recovery signals such as destination preference, regional characteristics, or media attention.

The goal is not prediction but interpretation. We rely on descriptive visualization, indexed baselines, geographic mapping, and text analytics to highlight structural patterns and identify any windows where spending and visitor activity diverge. Triangulating these different views allows us to assess whether the "tourism paradox" appears in the data or primarily in the stories told about the recovery.

Data Sources and Processing

California Transient Occupancy Tax Data

Joseph compiled fiscal-year lodging tax receipts for cities and counties across California from the State Controller's Office and supplemental municipal reports. The dataset includes general TOT revenues and annual totals for hundreds of jurisdictions, providing a broad administrative view of tourism-related economic activity in a large and diverse state.

Data processing involved standardizing fiscal-year formats (for example, 2018–19), converting TOT revenue fields to numeric types, and merging multiple source files into a consolidated dataset. Records with implausible totals or clearly miscoded entities were flagged and reviewed. City-level data were then aggregated to the county level for visual clarity. County shapefiles from official sources were cleaned and joined to the revenue data to support choropleth heatmaps and county-level bar charts.

New York State Visitor Spending Data

Hung used publicly available regional tourism data from the New York State tourism authorities, covering multiple tourism regions across the state. The dataset spans 2019–2024 and includes

annual visitor-spending totals for predefined regions such as New York City, Hudson Valley, Long Island, Greater Niagara, the Adirondacks, and others.

To enable proportional cross-year comparison, spending metrics were indexed to a 2019 baseline. For each region, a 2019 value of 100 represents pre-pandemic spending; subsequent years show percentage changes relative to that baseline. Year-over-year changes and recovery percentages were computed to track how quickly regions exceeded their 2019 levels. County boundaries from the *tigris* package were joined to region definitions to create a geographic map layer used in the dashboard.

National Media Article Dataset
Tanmoy compiled a corpus of tourism-related news articles from 2019 to 2025, resulting in 2,914 articles indexed through Google News. The dataset includes article text, metadata, keyword flags for state mentions and COVID references, sentiment scores, and topic-model assignments.

Text cleaning involved lowercasing, stripping punctuation, tokenization, and removal of standard stopwords. Additional custom stopwords (e.g., boilerplate website strings) were added as needed. A six-topic Latent Dirichlet Allocation (LDA) model was trained on the cleaned corpus to identify thematic clusters. Lexicon-based sentiment analysis produced both average sentiment scores and the share of articles classified as positive by year. For geographic emphasis, a dictionary of U.S. state names was used to count mentions across articles.

Visualization Tools

All three dashboards were built using R and Shiny. Core visualization packages included **ggplot2** for charts, often converted to interactive displays using **plotly**; **scales** for currency and percentage formatting; and **bslib** for theming. Geographic visualizations relied on **sf** for spatial data structures, **leaflet** for interactive choropleth maps, and **tigris** for official county and regional boundaries.

Data wrangling used **dplyr** and **tidyr** for cleaning and aggregation, **readxl** and **readr** for importing Excel and CSV files, **stringr** for text manipulation, and **reshape2** where reshaping was required. For the media narratives dashboard, additional text-analysis tools included **tidytext**, **stm** or **topicmodels** (depending on implementation), and **wordcloud** for visualizing frequent terms. A custom helper function ensured that missing files in the deployed app did not crash the dashboard but instead produced informative messages for the user.

Analytical Approach

Across the three dashboards, we emphasize four common analytical principles:

1. **Indexing and baselines.** Where possible, we normalize values to a 2019 baseline of 100. This allows direct comparison of recovery trajectories even when underlying levels differ dramatically.

2. **Geographic concentration.** We examine whether tourism spending is concentrated in a small number of regions or counties and how those focal areas behave relative to the rest.

3. **Temporal recovery patterns.** We compare pre-pandemic, pandemic, and post-pandemic years to see whether recovery is smooth, staggered, or uneven.

4. **Narrative alignment.** Finally, we assess whether media sentiment and topic emphasis reinforce, complicate, or contradict the patterns seen in the administrative and regional spending data.

Results

California Transient Occupancy Tax Patterns

The California TOT dashboard reveals substantial geographic concentration of tourism-related tax revenue. The county heatmap shows that coastal and major tourist counties generally have higher TOT values, while inland and rural counties collect significantly less. San Francisco, Los Angeles, Orange, San Diego, and Alameda consistently top the revenue rankings. Coastal tourist regions—particularly the Bay Area and Southern California coast—form a clear high-TOT cluster.

Over the course of the observed fiscal-year timeline, the Bay Area cluster lost ground relative to Southern California. By the end of the period, the three Southern California counties—Los Angeles, San Diego, and Orange - surpassed Bay Area counties in TOT levels, suggesting that sun-and-beach destinations may have been somewhat more resilient than dense urban cores built around business and international travel.

The county bar chart reinforces this pattern, showing a highly skewed distribution in which a small number of counties account for the majority of TOT revenue. Most counties have relatively low TOT in comparison. A few sparsely populated counties with resort destinations show moderate TOT spikes, implying that tourism intensity per resident varies considerably even when absolute revenue remains modest. Because only fiscal-year totals were available, the dashboard

cannot show intra-year seasonality, and any apparent “softness” in particular years may be driven by how fiscal years intersect with pandemic waves and reopenings.

New York Regional Spending Recovery

The New York dashboard compares visitor-spending recovery across multiple tourism regions. All regions show a strong rebound after the 2020 disruption, surpassing 2019 visitor-spending levels by 2022. Statewide, spending accelerates from 2021 through 2023, indicating a robust and fairly broad-based recovery.

However, recovery rates are not uniform. Long Island and the Hudson Valley recover more slowly than the state average, while regions with major urban or iconic natural attractions recover fastest. The Adirondacks—comprising nine upstate counties—exhibits especially rapid growth. By 2024, Adirondacks spending reaches 161.8% of its 2019 level, compared with 127.7% for New York State overall. The line chart visually emphasizes this divergence: both the state and the region dip sharply in 2020, but the Adirondacks’ index climbs more steeply from 2021 onward.

The region map highlights spatial clusters of tourism activity. Three of the four fastest-recovering regions—New York City, Hudson Valley, and Long Island—cluster around the New York City metropolitan area, while Greater Niagara stands out for its high visitor volumes tied to Niagara Falls. The 2024 bar chart and pie chart demonstrate that visitor spending is highly concentrated. New York City, Hudson Valley, Greater Niagara, and Long Island together generate more than half of statewide visitor spending, and New York City alone accounts for over 35% of annual spending in every year since 2019.

These top regions do not just return to baseline; they move ahead of the rest of the state by 2023 and 2024. At the same time, the Adirondacks’ strong relative recovery shows that smaller, outdoor-oriented regions can outperform the state overall even while representing only 2.65% of total spending in 2024. None of these patterns support a prolonged decoupling between visitor activity and spending; instead, they point to geographically uneven but broadly positive recovery.

Media Narratives and Sentiment Patterns

The media narratives dashboard analyzes 2,914 tourism-related articles from 2019 to 2025. The sentiment timeline shows a clear recovery in the tone of tourism reporting after the shock of early 2020. Coverage becomes increasingly positive beginning in 2021, with visible peaks in 2022 and again in 2024. Both the share of positive articles and average sentiment scores follow the same broad pattern: a sharp dip during the height of COVID-19, followed by sustained optimism as travel resumes and borders reopen.

Topic modeling yields six stable themes:

1. National parks and outdoor destinations,
2. Global tourism and markets,
3. Cities, beaches, and domestic destinations,
4. Medical tourism and safety,
5. International tourists and spending, and
6. Travel rules, border controls, and visas.

Top-term plots show that each topic is semantically coherent, and the presence of safety- and rules-oriented themes even after 2022 suggests that risk communication remains part of the tourism narrative beyond the immediate crisis.

State mentions in the corpus show Florida and California dominating media attention, consistent with their roles as major U.S. tourism economies. States with prominent national parks—such as Arizona, Utah, and Colorado—also rank highly, supporting the observation that outdoor recreation became a key recovery motif. The word cloud and frequency tables confirm that the most common words across the corpus include *travel*, *visitors*, *tourists*, and *recovery*, indicating that journalists frame tourism primarily through mobility and economic rebound rather than sustained pessimism.

Overall, the media data align more with the New York and California spending patterns than with the stronger versions of the "tourism paradox" narrative. Negative coverage is heavily concentrated in 2020; afterwards, tone becomes steadily more positive and remains so, especially in stories about outdoor destinations and domestic travel.

Discussion

Taken together, the three dashboards provide converging evidence that tourism recovery, when measured carefully, does not support a long-lasting "crowds without cash" pattern. In California, TOT revenues follow predictable geographic patterns concentrated in established tourist destinations. In New York, indexed spending shows that all regions surpass pre-pandemic levels by 2022, with urban centers and iconic attractions recovering most quickly. In national media coverage, sentiment tracks economic recovery closely, becoming more positive as destinations reopen and restrictions ease.

The apparent paradox—busy destinations with weak revenues—may instead reflect several measurement and framing issues. First, fiscal-year or annual aggregation can obscure seasonal patterns and short-term dips. A destination might experience a strong summer but a weak winter; in yearly totals, this nuance disappears. Second, the shift from international to domestic travelers during 2020–2021 temporarily reduced average spending per visitor, since domestic travelers typically spend less on lodging and long-haul flights. Third, inflation and currency effects matter: nominal revenue increases may mask real purchasing-power changes if not properly deflated. Finally, media narratives often emphasize crowding at particular destinations (e.g., national parks) without providing parallel context on tax receipts or spending indices.

A consistent finding across both California and New York is geographic concentration. In California, a handful of coastal counties generate the majority of TOT revenue. In New York, four regions—New York City, Hudson Valley, Greater Niagara, and Long Island—account for over half of statewide visitor spending. This concentration means that aggregate national or state-level statistics can easily obscure important variation in how different destination types perform. The media corpus partly reflects this concentration: Florida, California, and major gateway destinations receive disproportionate attention, while smaller regions—even those with strong relative recovery like the Adirondacks—rarely dominate national coverage.

The triangulation across administrative data, regional indices, and news narratives is crucial. If any single data source were taken in isolation, it would be easy to overstate anomalies or to misinterpret short-term divergence as a structural break. By comparing patterns across datasets, we find a more tempered story: tourism recovery has been uneven but broadly aligned across visitors, spending, and sentiment.

Limitations

Several limitations affect how our results should be interpreted.

For California, only fiscal-year TOT totals were available, preventing analysis of monthly or seasonal variations. Some cities and counties had missing or incomplete data, and differences in reporting conventions across jurisdictions (for example, whether certain fees are included in TOT or classified under other revenue categories) limit strict comparability. Trend interpretation around fiscal year 2020–21 relies on partially missing or estimated values, which introduces uncertainty.

For New York, data from the state tourism authorities are annual rather than monthly, limiting temporal granularity. The spending totals are not broken down by category, such as lodging, retail, or dining, so we cannot determine which sectors recovered fastest. The unusual pattern observed in the Hudson Valley region—possibly influenced by real-estate transactions or second-home

markets—cannot be explored further due to lack of categorical detail. Region definitions are predetermined by the state, which simplifies mapping but may mask sub-regional differences.

For the media analysis, PDF extraction occasionally produced noisy text and formatting artifacts. Metadata gaps prevented more detailed geographic coding beyond state mentions, and topic labels are inherently researcher-driven, as with any unsupervised model. In some years, the number of tourism articles is modest, making sentiment averages sensitive to outlier stories. Because the corpus is drawn from English-language articles indexed by Google News, it reflects a U.S.-centric digital media environment and cannot be treated as a global sample of tourism narratives.

Finally, automated sentiment analysis and topic modeling necessarily simplify complex stories. Articles often combine positive and negative elements, quote multiple actors, or frame uncertainty in nuanced ways that a numeric score cannot fully capture. We therefore treat sentiment and topic outputs as structured signals rather than definitive judgments about individual articles.

Conclusion

This project demonstrates how combining different forms of evidence—administrative tax records, regional spending indices, and media narratives—can clarify whether tourism recovery patterns match public perception. Our dashboards suggest that the "tourism paradox" is largely a measurement and framing artifact rather than robust evidence of a long-term disconnect between visitors and spending. When data are cleaned, indexed, and visualized transparently, spending and visitor activity tend to move together in ways that align with standard expectations.

The project also illustrates the value of interactive visualization for policy-relevant analysis. Shiny dashboards allow users to explore patterns across time, geography, and topic, making complex data more accessible to non-technical audiences. For tourism planners and destination-marketing organizations, tools like these provide a template for monitoring recovery using public data and for checking whether widely shared narratives match the underlying numbers.

Future work could extend this framework by incorporating monthly or even weekly data where available, adding visitor-mobility indicators from transportation sources, and comparing U.S. patterns with those in other countries. The methods developed here—indexed baselines, geographic clustering, and narrative alignment—can be adapted to other policy domains where researchers need to understand how economic behavior and public discourse interact.

References

Bureau of Labor Statistics. (2025). Consumer Price Index for All Urban Consumers (CPI-U), U.S. city average [Data set]. U.S. Department of Labor.
https://www.bls.gov/cpi/

Bureau of Transportation Statistics. (2025). Air Carrier Statistics (T-100), domestic market and international segment [Data set]. U.S. Department of Transportation.
https://www.transtats.bts.gov/

California Department of Tax and Fee Administration. (2025). Transient Occupancy Tax (TOT) statistics [Data set].
https://www.cdtfa.ca.gov/taxes-and-fees/tot-program.htm

OECD. (2024). OECD tourism trends and policies 2024. OECD Publishing.
https://doi.org/10.1787/80885d8b-en

Papagianni, E., Evgenidis, A., Tsagkanos, A., & Megalooikonomou, V. (2023). Tourism demand in the face of geopolitical risk: Insights from a cross-country analysis. Journal of Travel Research, 63(8), 2094–2119.
https://doi.org/10.1177/00472875231206539

Sampaio, C., Sebastião, J. R., & Farinha, L. (2024). Hospitality and tourism demand: Exploring industry shifts, themes, and trends. Societies, 14(10), Article 207.
https://doi.org/10.3390/soc14100207

UN Tourism. (2025, January 21). International tourism recovers pre-pandemic levels in 2024 [Press release].
https://www.unwto.org/news/international-tourism-recovers-pre-pandemic-levels-in-2024

Appendix A

AI Use Documentation

This project used generative AI tools to support dashboard development, preparation of explanatory text, and refinement of the written report. No AI system generated data, performed statistical calculations, or made analytic decisions on behalf of the researchers; all final judgments and interpretations were made by the students.

Purpose of AI use.

AI assistance was used for the following purposes:

1. **Clarifying methodological steps**. AI helped refine workflows for loading, cleaning, and organizing data; interpreting model outputs; and identifying appropriate visual encodings.
2. **Debugging and troubleshooting.** When Shiny apps produced deployment errors, AI was used to identify likely causes, such as file-path inconsistencies and missing packages, and to suggest corrective steps.
3. **Improving clarity of written communication.** AI tools helped polish phrasing in results summaries and narrative descriptions to ensure clear, readable, and academically appropriate language.
4. **Brainstorming visualization strategies.** AI provided ideas about color choices, dashboard layout, and usability improvements but did not generate the charts themselves.
5. **Non-substantive editing.** AI-supported grammar fixes, minor restructuring, and alignment with APA stylistic norms.

Tools used.

The primary AI tools used were ChatGPT (GPT-5.1) and Claude (Claude Opus 4.5), accessed through web interfaces. Contributions included reviewing the conceptual structure of topic modeling, sentiment scoring, and geographic aggregation; helping refine R code for visualization and Shiny deployment; providing explanations that clarified ambiguous results; and drafting preliminary versions of some narrative paragraphs, which were then edited and verified by the authors. AI was not used for statistical decision-making, automated web scraping, or running the analyses themselves.

Appendix B

Code Files and Contributions
Below is a description of the main code files used in the project and the team member responsible for each dashboard. The full code appears on separate pages following this appendix.

Tanmoy Biswas — Tourism News Text Analysis Dashboard

- **app.R** – Main Shiny application for the media narratives dashboard. Handles UI layout (controls, metric cards, topic-terms plot, sentiment timeline, state-mentions bar chart, word cloud, and article browser) and server logic. Includes:
    - Safe loading of CSV files for topics, article metadata, state mentions, top words, and sentiment timeline.
    - Helper functions for counting indicator variables and formatting metrics.
    - Plotly-based visualizations for topic-term bar charts, sentiment trajectories, and state-mention counts.
    - A word cloud of top terms using viridis colors.
    - A cleaned article browser table using DT with Yes/No indicators for state and COVID mentions.

Additional supporting scripts (if included with the submission) cover data cleaning, topic modeling, sentiment scoring, and state-mention extraction prior to exporting the CSV files used by the app.

The full code begins on page 13.

Joseph Martinez — California TOT Dashboard

R files include:

- Data-cleaning script for reading Excel sheets of California TOT data, standardizing city names, mapping cities to counties using spatial joins, and aggregating revenues to the county level.
- Shiny app file defining the UI (fiscal-year selector, view options for heatmap, bar chart, and data table) and server logic.
- Geographic plotting code using sf and tigris to render county-level heatmaps and a horizontal bar chart of TOT revenue by county.

The full code begins on page 28

Hung Tran — New York Tourism Recovery Dashboard

R files include:

- Scripts to load regional tourism data from Excel, reshape it to long format, define official tourism regions, and compute 2019-baseline spending indices.
- Shiny app file implementing the controls (region selector, year-range slider, and focus-year dropdown) and the main UI layout.
- Visualization code that:
    - Plots region vs. statewide spending indices over time.
    - Generates bar charts and pie charts for spending levels and shares in a selected year.
    - Builds an interactive leaflet choropleth map of the selected region's counties using sf and tigris.
    - Produces a dynamic "recovery summary" text box that reports current recovery percentages and the first year a region returned to or exceeded its 2019 level.

The full code begins on page 21

Full Code, Tanmoy

```
# ============================================================
# Tourism News Text Analysis Dashboard
# EPPS 6356 - Data Visualization (Fall 2025)
# app.R  (place in: shiny_app/app.R)
# ============================================================

library(shiny)
library(bslib)
library(tidyverse)
library(readr)
library(plotly)
library(DT)
library(viridis)
library(wordcloud)

# ---- Safe loader ------------------------------------------

safe_read_csv <- function(path) {
  if (file.exists(path)) read_csv(path, show_col_types = FALSE) else NULL
}

# NOTE: paths are relative to the app directory on the server
topics_results     <- safe_read_csv("data/topics_results.csv")
article_data       <- safe_read_csv("data/article_metadata.csv")
state_mentions     <- safe_read_csv("data/state_mentions.csv")
top_words          <- safe_read_csv("data/top_words.csv")
sentiment_timeline <- safe_read_csv("data/sentiment_timeline.csv")

# ---- Topic labels (for dropdown) --------------------------

topic_labels <- c(
  "1" = "National parks",
  "2" = "Global tourism & markets",
  "3" = "Cities, beaches & destinations",
  "4" = "Medical tourism & safety",
  "5" = "International tourists & spending",
  "6" = "Travel rules & visas"
)

# ---- Prepare sentiment data -------------------------------

sentiment_df <- NULL
if (!is.null(sentiment_timeline)) {
  # file has: time_period, n_articles, avg_sentiment, pct_positive
  sentiment_df <- sentiment_timeline |>
    mutate(
      year = as.integer(time_period),
      date = as.Date(paste0(year, "-01-01"))
    )
}

# ---- Colors / theme ---------------------------------------

pal_florida    <- "#4169E1"  # blue
pal_california <- "#FF8C00"  # orange
pal_neutral    <- "#6C757D"  # grey
```

```
app_theme <- bs_theme(
  bootswatch = "flatly",
  base_font  = font_google("Source Sans 3")
)

# ---- Helper: safely count indicator columns ----------

indicator_count <- function(df, indicator_name) {
  if (is.null(df)) return(NA_integer_)
  nms <- names(df)
  # case-insensitive column match
  hit <- nms[tolower(nms) == tolower(indicator_name)]
  if (length(hit) == 1) {
    v <- df[[hit]]
    if (is.logical(v) || is.numeric(v)) {
      return(sum(v, na.rm = TRUE))
    }
  }
  NA_integer_
}

# ---- UI ----------------------------------------------------

ui <- fluidPage(
  theme = app_theme,
  titlePanel("Tourism Recovery: Media Narratives"),

  # Analysis period note just under the title
  div(
    style = "margin-bottom: 15px; font-size: 14px; font-weight: 500;",
    "Analysis period: 2019–2025 news coverage on tourism recovery"
  ),

  sidebarLayout(
    sidebarPanel(
      width = 3,
      h4("Controls"),

      selectInput(
        "topic_select",
        "Choose topic:",
        choices = if (!is.null(topics_results) && "topic" %in% names(topics_results)) {
          topic_ids <- sort(unique(topics_results$topic))
          setNames(topic_ids, topic_labels[as.character(topic_ids)])
        } else NULL
      ),

      sliderInput(
        "top_terms_n",
        "Number of top terms per topic:",
        min = 3, max = 10, value = 5
      ),

      checkboxInput(
        "use_pct_positive",
        "Sentiment timeline: show % positive (instead of average score)",
        value = FALSE
      ),
```

```
  hr(),
  helpText(
    "Tip: hover over bars and lines for exact values; ",
    "use the article browser to inspect individual stories."
  )
),

mainPanel(
  width = 9,

  # ---- Metric cards ----
  fluidRow(
    column(
      2,
      wellPanel(
        style = "min-height: 130px;",
        h5("Corpus Size"),
        h3(
          textOutput("ta_total_articles"),
          style = "font-weight:700; font-size:26px; color:#4169E1;"
        ),
        p("Total articles", style = "font-size: 11px;")
      )
    ),
    column(
      2,
      wellPanel(
        style = "min-height: 130px;",
        h5("Florida Coverage"),
        h3(
          textOutput("ta_fl_articles"),
          style = "font-weight:700; font-size:26px; color:#4169E1;"
        ),
        p("Articles mentioning FL", style = "font-size: 11px;")
      )
    ),
    column(
      2,
      wellPanel(
        style = "min-height: 130px;",
        h5("California Coverage"),
        h3(
          textOutput("ta_ca_articles"),
          style = "font-weight:700; font-size:26px; color:#4169E1;"
        ),
        p("Articles mentioning CA", style = "font-size: 11px;")
      )
    ),
    column(
      3,
      wellPanel(
        style = "min-height: 130px;",
        h5("COVID Coverage"),
        h3(
          textOutput("ta_covid_articles"),
          style = "font-weight:700; font-size:26px; color:#4169E1;"
        ),
        p("Articles mentioning COVID-19", style = "font-size: 11px;")
      )
```

```
      ),
      column(
        3,
        wellPanel(
          style = "min-height: 130px;",
          h5("Most Common Word"),
          h3(
            textOutput("ta_top_word"),
            style = "font-weight:700; font-size:26px; color:#4169E1;"
          ),
          p("Across entire corpus", style = "font-size: 11px;")
        )
      )
    ),

    br(),

    # ---- Topic terms & sentiment ----
    fluidRow(
      column(
        6,
        h4("Topic Modeling: Top Terms"),
        plotlyOutput("ta_topics_plot", height = "420px")
      ),
      column(
        6,
        h4("Sentiment Timeline"),
        plotlyOutput("ta_sentiment_plot", height = "420px")
      )
    ),

    br(),

    # ---- States & word cloud ----
    fluidRow(
      column(
        6,
        h4("Top ten states mentioned in articles"),
        plotlyOutput("ta_states_plot", height = "380px")
      ),
      column(
        6,
        h4("Top Words"),
        plotOutput("ta_wordcloud", height = "380px")
      )
    ),

    br(),

    # ---- Article table ----
    h4("Article Browser"),
    DTOutput("ta_articles_table")
  )
 )
)

# ---- Server ------------------------------------------------

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

```
# -------- Metric cards --------

output$ta_total_articles <- renderText({
  if (is.null(article_data)) return("N/A")
  format(nrow(article_data), big.mark = ",")
})

output$ta_fl_articles <- renderText({
  n <- indicator_count(article_data, "mentions_florida")
  if (is.na(n)) "N/A" else format(n, big.mark = ",")
})

output$ta_ca_articles <- renderText({
  n <- indicator_count(article_data, "mentions_california")
  if (is.na(n)) "N/A" else format(n, big.mark = ",")
})

output$ta_covid_articles <- renderText({
  n <- indicator_count(article_data, "mentions_covid")
  if (is.na(n)) "N/A" else format(n, big.mark = ",")
})

output$ta_top_word <- renderText({
  if (!is.null(top_words) && all(c("word", "n") %in% names(top_words))) {
    top_words |>
      arrange(desc(n)) |>
      slice(1) |>
      pull(word)
  } else {
    "\"travel\""
  }
})

# -------- Topic terms plot --------

output$ta_topics_plot <- renderPlotly({
  req(topics_results, input$topic_select)

  df <- topics_results
  validate(
    need(all(c("topic", "term", "beta") %in% names(df)),
         "Topic results file is missing required columns.")
  )

  df_topic <- df |>
    filter(topic == input$topic_select) |>
    arrange(desc(beta)) |>
    slice_head(n = input$top_terms_n) |>
    mutate(term = forcats::fct_reorder(term, beta))

  topic_name <- topic_labels[as.character(input$topic_select)]

  p <- ggplot(df_topic, aes(x = beta, y = term)) +
    geom_col(fill = pal_florida) +
    theme_minimal(base_size = 13) +
    labs(
      x = "Term importance (beta)",
      y = NULL,
```

```
      title = paste("Top terms for:", topic_name)
    )

  ggplotly(p, tooltip = c("y", "x")) |>
    layout(margin = list(t = 70))
})

# -------- Sentiment timeline --------

output$ta_sentiment_plot <- renderPlotly({
  req(sentiment_df)

  df <- sentiment_df
  validate(
    need(all(c("date", "avg_sentiment", "pct_positive") %in% names(df)),
         "sentiment_timeline.csv must include time_period/avg_sentiment/pct_positive.")
  )

  if (isTRUE(input$use_pct_positive)) {
    df_plot <- df |>
      mutate(value = pct_positive / 100) |>  # convert to 0–1
      select(date, value)

    y_lab  <- "% positive articles"
    y_fmt  <- scales::label_percent(accuracy = 1)
    baseline <- 0.5   # 50% threshold
  } else {
    df_plot <- df |>
      select(date, value = avg_sentiment)

    y_lab  <- "Average sentiment score"
    y_fmt  <- scales::label_number(accuracy = 1)
    baseline <- 0
  }

  p <- ggplot(df_plot, aes(x = date, y = value)) +
    geom_line(color = pal_florida, linewidth = 1.2) +
    geom_point(color = pal_florida, size = 2) +
    geom_hline(yintercept = baseline,
               linetype = "dashed", color = pal_neutral) +
    scale_y_continuous(labels = y_fmt) +
    scale_x_date(date_labels = "%Y") +
    theme_minimal(base_size = 14) +
    theme(
      panel.grid.minor = element_blank(),
      plot.margin = margin(t = 10, r = 10, b = 10, l = 10)
    ) +
    labs(x = NULL, y = y_lab)

  ggplotly(p) |>
    layout(margin = list(t = 50))
})

# -------- State mentions --------

output$ta_states_plot <- renderPlotly({
  req(state_mentions)

  df <- state_mentions
```

```
  validate(
    need(all(c("state", "count") %in% names(df)),
         "state_mentions.csv needs 'state' and 'count' columns.")
  )

  df <- df |>
    slice_max(order_by = count, n = 10) |>
    mutate(state = forcats::fct_reorder(state, count))

  p <- ggplot(df, aes(x = count, y = state)) +
    geom_col(fill = pal_florida) +
    theme_minimal(base_size = 14) +
    theme(
      plot.title = element_text(size = 14, face = "bold")
    ) +
    labs(
      title = "Top ten states mentioned in articles",
      x = "Number of articles",
      y = NULL
    )

  ggplotly(p) |>
    layout(margin = list(t = 70))
})

# -------- Word cloud --------

output$ta_wordcloud <- renderPlot({
  req(top_words)

  df <- top_words
  validate(
    need(all(c("word", "n") %in% names(df)),
         "top_words.csv needs 'word' and 'n' columns.")
  )

  df <- df |>
    filter(!word %in% c("news.google.com")) |>
    arrange(desc(n)) |>
    slice_head(n = 80)

  par(mar = c(0, 0, 0, 0))
  wordcloud(
    words        = df$word,
    freq         = df$n,
    max.words    = 80,
    colors       = viridis(8),
    random.order = FALSE,
    scale        = c(3.2, 0.8)
  )
})

# -------- Article browser --------

output$ta_articles_table <- renderDT({
  req(article_data)

  cols <- intersect(
    c(
```

```
      "article_id", "filename", "title", "time_period",
      "mentions_florida", "mentions_california",
      "mentions_covid", "positive", "negative",
      "sentiment_score"
    ),
    names(article_data)
  )

  df <- article_data[, cols, drop = FALSE]

  # Convert logicals to Yes/No for display
  logical_cols <- c("mentions_florida", "mentions_california",
                    "mentions_covid", "positive", "negative")

  for (col in logical_cols) {
    if (col %in% names(df)) {
      df[[col]] <- ifelse(df[[col]] == TRUE, "Yes",
                          ifelse(df[[col]] == FALSE, "No", df[[col]]))
    }
  }

  nice_names <- c(
    article_id         = "ID",
    filename           = "File",
    title              = "Title",
    time_period        = "Year",
    mentions_florida   = "Mentions Florida",
    mentions_california = "Mentions California",
    mentions_covid     = "Mentions COVID",
    positive           = "Positive",
    negative           = "Negative",
    sentiment_score    = "Sentiment score"
  )
  colnames(df) <- nice_names[colnames(df)]

  datatable(
    df,
    options  = list(
      pageLength = 25,
      autoWidth  = TRUE,
      searching  = FALSE,   # no global search box
      dom        = "tip"    # table + information + pagination
    ),
    rownames = FALSE
  )
 })
}

shinyApp(ui, server)
```

Full Code, Hung

```
library(shiny)
library(readxl)
library(dplyr)
library(tidyr)
library(ggplot2)
library(plotly)
library(sf)
library(tigris)
library(leaflet)
library(stringr)
library(scales)

options(tigris_use_cache = TRUE)
# ------------------------------------------------------------------
# 1. LOAD EXCEL FILES
# ------------------------------------------------------------------
excel_path <- "NewYorkState.xlsx"   # adjust if needed

# --- Regions sheet (for county → region map) ----
regions_df <- read_excel(excel_path, sheet = "Regions")

# Split comma-separated county lists and clean names
region_counties <- regions_df |>
  mutate(Counties = str_split(Counties, ",\\s*")) |>
  unnest(Counties) |>
  mutate(
    County = str_trim(Counties),
    # remove things in parentheses: "(part)", "(Brooklyn)", etc.
    County = str_replace(County, " \\([^)]*\\)", "")
  ) |>
  select(Region, County)

# --- State - Visitor Spending sheet (region + state totals) ----
state_sp_raw <- read_excel(excel_path, sheet = "State - Visitor Spending")

# First column is "Visitor Spending ($ millions)" – rename it to Name
state_long <- state_sp_raw |>
  rename(Name = 1) |>
  pivot_longer(
    cols      = -Name,
    names_to  = "Year",
    values_to = "Spending"
  ) |>
  mutate(
    Year = as.integer(Year),
    Region = case_when(
      Name == "State" ~ "New York State",
      TRUE ~ str_trim(str_remove(Name, "^[0-9]+\\.\\s*"))
    )  )

# Keep only the 11 tourism regions + state row
valid_regions <- c(regions_df$Region, "New York State")
tourism <- state_long |>
  filter(Region %in% valid_regions)

# ------------------------------------------------------------------
# 2. 2019 BASELINE INDEX
```

```
# ------------------------------------------------------------------
baseline <- tourism |>
  filter(Year == 2019) |>
  select(Region, Spending_2019 = Spending)

tourism_index <- tourism |>
  left_join(baseline, by = "Region") |>
  mutate(
    SpendingIndex = 100 * Spending / Spending_2019
  )

year_min <- max(2019, min(tourism_index$Year, na.rm = TRUE))
year_max <- max(tourism_index$Year, na.rm = TRUE)

# ------------------------------------------------------------------
# 3. NY COUNTY GEOMETRY + JOIN TO REGIONS
# ------------------------------------------------------------------
ny_counties <- counties(state = "NY", cb = TRUE, year = 2022) |>
  st_as_sf() |>
  mutate(NAME = str_trim(NAME))

region_counties_sf <- region_counties |>
  left_join(ny_counties, by = c("County" = "NAME")) |>
  st_as_sf()

# ------------------------------------------------------------------
# 4. UI
# ------------------------------------------------------------------
ui <- fluidPage(
  # --- styles ---
  tags$head(
    tags$style(HTML("
      .summary-box {
        padding: 16px;
        border: 1px solid #ccc;
        border-radius: 8px;
        background: #fff;
        margin-top: 16px;
        box-shadow: 0 2px 6px rgba(0,0,0,0.05);
      }
      #summary_text {
        white-space: normal;
        word-wrap: break-word;
        overflow-wrap: break-word;
        font-size: 15px;
        line-height: 1.5;
      }
    ")) ),

  # ---- TITLE ----
  titlePanel("New York Tourism Recovery"),

  # ---- CONTROLS + CROPPED IMAGE ROW ----
  fluidRow(
    column(
      width = 4,
      # fixed-height container so it aligns with image
      div(
        style = "height:260px;",
```

```
    wellPanel(
      style = "height:100%; overflow-y:auto;",
      selectInput("region", "Choose region:",
                  choices = sort(unique(regions_df$Region))),
      sliderInput("year_range", "Year range:",
                  min = year_min, max = year_max,
                  value = c(year_min, year_max),
                  step = 1, sep = ""),
      selectInput("year_focus", "Year for bar & pie charts:",
                  choices = seq(year_min, year_max),
                  selected = year_max)
    ) ) ),
  column(
    width = 8,
    # same height, cropped image aligned with sidebar
    div(
      style = "
        height:260px;
        overflow:hidden;
        border-radius:8px;
        border:1px solid #ccc;
        box-shadow:0 2px 5px rgba(0,0,0,0.1);
      ",
      tags$img(
        src   = "nyc.jpg",       # file should be in ./www/nyc.jpg
        style = "
          width:100%;
          height:100%;
          object-fit:cover;      /* crop to fill box */
          display:block;
        "
      ) ) ) ),

br(),

# ---- MAP + CHARTS/SUMMARY ROW ----
fluidRow(
  # left: map
  column(
    width = 4,
    h4("Region map"),
    leafletOutput("region_map", height = 520)
  ),

  # right: charts + summary
  column(
    width = 8,
    div(
      style = "margin-top:30px;",

      fluidRow(
        column(width = 4, plotlyOutput("line_index", height = 260)),
        column(width = 4, plotlyOutput("bar_level",  height = 260)),
        column(width = 4, plotlyOutput("pie_share",   height = 260))
      ),

      div(
        class = "summary-box",
        h4("Recovery summary"),
```

```
        textOutput("summary_text")
      ) ) ) ) )
# ------------------------------------------------------------------
# 5. SERVER
# ------------------------------------------------------------------
server <- function(input, output, session) {

 # Keep year_focus inside selected range
 observe({
  yrs <- seq(input$year_range[1], input$year_range[2])
  updateSelectInput(
   session,
   "year_focus",
   choices  = yrs,
   selected = max(yrs)
  )
 })

 # Region time series
 region_data <- reactive({
  tourism_index |>
   filter(
    Region == input$region,
    Year >= input$year_range[1],
    Year <= input$year_range[2]
   ) |>
   arrange(Year)
 })

 # State time series
 state_data <- reactive({
  tourism_index |>
   filter(
    Region == "New York State",
    Year >= input$year_range[1],
    Year <= input$year_range[2]
   ) |>
   arrange(Year)
 })

 # Region vs state in focus year
 focus_data <- reactive({
  tourism_index |>
   filter(
    Year   == input$year_focus,
    Region %in% c(input$region, "New York State")
   )
 })

 # ---- LINE: RECOVERY INDEX (2019 = 100) ----
 output$line_index <- renderPlotly({
  reg <- region_data()
  st  <- state_data()
  req(nrow(reg) > 0, nrow(st) > 0)

  p <- ggplot() +
   geom_hline(yintercept = 100, linetype = "dotted", color = "grey40") +
   geom_line(
    data = st,
```

```
      aes(x = Year, y = SpendingIndex, group = 1),
      linetype = "dashed"
    ) +
    geom_point(
      data = st,
      aes(x = Year, y = SpendingIndex)
    ) +
    geom_line(
      data = reg,
      aes(x = Year, y = SpendingIndex, group = 1),
      linewidth = 1.1
    ) +
    geom_point(
      data = reg,
      aes(x = Year, y = SpendingIndex)
    ) +
    scale_y_continuous("Visitor spending (2019 = 100)") +
    scale_x_continuous(breaks = seq(year_min, year_max, 1)) +
    labs(
      title = paste(input$region, "vs NY State"),
      x     = "Year"
    ) +
    theme_minimal()

  ggplotly(p)
})

# ---- BAR: SPENDING LEVELS IN FOCUS YEAR ----
output$bar_level <- renderPlotly({
  df <- focus_data()
  req(nrow(df) > 0)

  p <- ggplot(df, aes(x = Region, y = Spending, fill = Region)) +
    geom_col() +
    scale_y_continuous("Visitor spending ($ millions)", labels = comma) +
    labs(
      title = paste("Visitor spending in", input$year_focus),
      x     = ""
    ) +
    theme_minimal() +
    theme(legend.position = "none")

  ggplotly(p)
})

# ---- PIE: REGION SHARE OF STATE ----
output$pie_share <- renderPlotly({
  df <- focus_data()
  req(nrow(df) == 2)   # region + state

  reg_val <- df$Spending[df$Region == input$region]
  st_val  <- df$Spending[df$Region == "New York State"]

  share_df <- data.frame(
    Category = c(input$region, "Rest of State"),
    Value    = c(reg_val, st_val - reg_val)
  )

  plot_ly(
```

```
    data   = share_df,
    labels = ~Category,
    values = ~Value,
    type   = "pie"
  ) |>
    layout(
      title      = paste("Visitor Spending (%)", input$year_focus),
      legend     = list(orientation = "h", x = 0.1, y = -0.1),
      margin     = list(l = 0, r = 0, b = 40, t = 40)
    )
})


# ---- TEXT SUMMARY: HOW FAST & HOW MUCH RECOVERY ----
output$summary_text <- renderText({
  dat <- region_data()
  req(nrow(dat) > 0)

  latest <- dat |> slice(n())
  pct_latest <- round(latest$SpendingIndex, 1)

  # first year (>= 2020) where region reached or exceeded 2019 level
  first_full <- dat |>
    filter(Year >= 2020, SpendingIndex >= 100) |>
    slice(1)

  st_latest <- state_data() |>
    filter(Year == latest$Year)

  txt <- paste0(
    input$region, " is at ",
    pct_latest, "% of its 2019 visitor spending level as of ",
    latest$Year, ". "
  )

  if (nrow(first_full) > 0) {
    txt <- paste0(
      txt,
      "It first reached or exceeded its 2019 level in ",
      first_full$Year, ". "
    )
  } else {
    txt <- paste0(
      txt,
      "It has not yet fully returned to its 2019 level in the selected period. "
    )
  }

  if (nrow(st_latest) == 1 && !is.na(st_latest$SpendingIndex)) {
    state_pct <- round(st_latest$SpendingIndex, 1)
    txt <- paste0(
      txt,
      "For comparison, New York State overall is at ",
      state_pct, "% of its 2019 level in ",
      latest$Year, "."
    )
  }

  txt
```

```
  })

  # ---- MAP: REGION COUNTIES COLORED ----
  region_shape <- reactive({
    region_counties_sf |>
      filter(Region == input$region)
  })

  output$region_map <- renderLeaflet({
    shp <- region_shape()
    req(nrow(shp) > 0)

    pal <- colorFactor("Set3", domain = shp$County)

    bbox <- st_bbox(shp)
    center_lng <- as.numeric((bbox["xmin"] + bbox["xmax"]) / 2)
    center_lat <- as.numeric((bbox["ymin"] + bbox["ymax"]) / 2)

    leaflet(shp) |>
      addProviderTiles("CartoDB.Positron") |>
      addPolygons(
        fillColor   = ~pal(County),
        weight      = 1,
        opacity     = 1,
        color       = "white",
        fillOpacity = 0.7,
        label       = ~County,
        highlight   = highlightOptions(
          weight       = 2,
          color        = "#666",
          fillOpacity  = 0.8,
          bringToFront = TRUE
        )
      ) |>
      addLegend(
        "bottomright",
        pal     = pal,
        values  = ~County,
        title   = "Counties",
        opacity = 1
      ) |>
      setView(lng = center_lng, lat = center_lat, zoom = 7)
  })
}
shinyApp(ui, server)
```

Full Code, Joseph

```
library(shiny)
library(readxl)
library(DT)
library(tigris)
library(sf)
library(dplyr)
library(ggplot2)

# Define UI
ui <- fluidPage(
 theme = bslib::bs_theme(bootswatch = "lux"),
 titlePanel("California TOT Data Explorer"),

 sidebarLayout(
  sidebarPanel(
   # Add image at the top of sidebar
   img(src = "LASkyline.jpg", width = "100%", style = "margin-bottom: 20px;"),

   selectInput(
    inputId = "sheet_select",
    label = "Select Fiscal Year:",
    choices = c(
     "CalReport by Cities 2017-2018",
     "CalReport by Cities 2018-2019",
     "CalReport by Cities 2019-2020",
     "CalReport by Cities 2020-2021",
     "CalReport by Cities 2021-2022"
    ),
    selected = "CalReport by Cities 2021-2022"
   ),
   hr(),
   h4("View Options:"),
   checkboxInput("show_heatmap", "Show County Heatmap", value = TRUE),
   checkboxInput("show_barchart", "Show County Bar Chart", value = TRUE),
   checkboxInput("show_table", "Show Data Table", value = TRUE),
   width = 3
  ),

  mainPanel(
   h3(textOutput("sheet_title")),

   conditionalPanel(
    condition = "input.show_heatmap",
    h4("County Heatmap"),
    plotOutput("heatmap", height = "600px")
   ),

   conditionalPanel(
   condition = "input.show_barchart",
    h4("County TOT Bar Chart"),
    plotOutput("barchart", height = "600px"),
    br()
   ),

   # Conditional panels based on checkboxes
   conditionalPanel(
    condition = "input.show_table",
```

```
      h4("Data Table"),
      DTOutput("data_table"),
      br()
    ),

    width = 9
  )
 )
)

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

 # Cache tigris data
 options(tigris_use_cache = TRUE)

 # Load CA county geometry (load once)
 ca_counties <- reactive({
  counties(state = "CA", cb = TRUE, class = "sf") %>%
   select(NAME, geometry) %>%
   rename(County = NAME)
 })

 # Function: Map cities to counties and aggregate
 aggregate_cities_to_counties <- function(df, tax_col) {
  colnames(df)[1] <- "City"
  df <- df %>% mutate(City = trimws(City))

  df <- df %>% filter(!is.na(City), !grepl("total|statewide", City, ignore.case = TRUE))
  df[[tax_col]] <- as.numeric(df[[tax_col]])

  ca_places <- places(state = "CA", cb = TRUE, class = "sf")
  ca_counties_full <- counties(state = "CA", cb = TRUE, class = "sf")

  place_centroids <- st_centroid(ca_places)
  place_county_map <- st_join(place_centroids, ca_counties_full) %>%
   st_drop_geometry() %>%
   select(place_name = NAME.x, county_name = NAME.y) %>%
   distinct()

  df <- df %>%
   mutate(
    City_clean = gsub(" city| town| City| Town", "", City, ignore.case = TRUE),
    City_clean = trimws(City_clean)
   )

  df_mapped <- df %>%
   left_join(place_county_map, by = c("City_clean" = "place_name"))

  df_aggregated <- df_mapped %>%
   filter(!is.na(county_name)) %>%
   group_by(County = county_name) %>%
   summarise(!!tax_col := sum(.data[[tax_col]], na.rm = TRUE), .groups = "drop")

  return(df_aggregated)
 }

 # Read the selected sheet
 sheet_data <- reactive({
```

```
  req(input$sheet_select)
  read_excel("taxes_cal.xlsx", sheet = input$sheet_select)
})

# Process data for heatmap
county_data <- reactive({
  req(sheet_data())
  df <- sheet_data()

  # Find tax column
  tax_col <- colnames(df)[grep("TransientOccupancy|Total Revenues", colnames(df), ignore.case = TRUE)][1]

  if (is.na(tax_col) || length(tax_col) == 0) {
    return(NULL)
  }

  # Aggregate cities to counties
  df_aggregated <- aggregate_cities_to_counties(df, tax_col)

  list(data = df_aggregated, tax_col = tax_col)
})

# Output the sheet title
output$sheet_title <- renderText({
  paste("California TOT Data  - ", input$sheet_select)
})
# Output the heatmap
output$heatmap <- renderPlot({
  req(county_data())

  county_info <- county_data()
  if (is.null(county_info)) return(NULL)

  df <- county_info$data
  tax_col <- county_info$tax_col

  # Join with map data
  map_data <- ca_counties() %>%
    left_join(df, by = "County")

  # Create plot
  ggplot(map_data) +
    geom_sf(aes(fill = .data[[tax_col]]), color = "white", size = 0.2) +
    scale_fill_viridis_c(option = "plasma", na.value = "grey90") +
    labs(
      title = paste("California TOT Heat Map  - ", input$sheet_select),
      fill = "TOT ($)"
    ) +
    theme_minimal() +
    theme(
      plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
    )
})

# Output Bar Chart
output$barchart <- renderPlot({
  req(county_data())

  df <- county_data()$data
```

```
    tax_col <- county_data()$tax_col

    df %>%
      ggplot(aes(x = reorder(County, .data[[tax_col]]), y = .data[[tax_col]])) +
      geom_col(fill = "#4C72B0") +
      coord_flip() +
      labs(
        title = "TOT Revenue by County",
        x = "County",
        y = "TOT ($)"
      ) +
      theme_minimal()
  })

  # Output the data table
  output$data_table <- renderDT({
    datatable(
      sheet_data(),
      options = list(
        pageLength = 10,
        autoWidth = TRUE,
        scrollX = TRUE
      ),
      rownames = FALSE
    )
  })

}

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