Assignment 2 - Reading On-line Data and Visualizing Hurricane Tracks

EVR-5086 Fall 2025

Adyan Rios

2025-09-16

Table of contents

Assignment 2 - Reading On-line Data and Visualizing Hurricane Tracks

1 Data Retrieval and Parsing

```
# Check if libraries are installed; install if not.
if (!require("pacman")) install.packages("pacman")
pacman::p_load(here, curl, tidyr, dplyr, lubridate)
```

```
# Specify the source and output file name and location
url <- "https://www.nhc.noaa.gov/data/hurdat/hurdat2-1851-2024-040425.txt"
destfile <- here("assignment2","hurdat.txt")

# Download and save the dataset
curl_download(url = url, destfile = destfile)
```

```r
# Read in data and differentiate between headers and data lines
lines <- readLines("hurdat.txt")                      # Read text file

# Prep headers
storm_headers <- lines[grepl("^AL", lines)]           # Keep only storm headers
header_parts <- strsplit(storm_headers, ",")          # Split based on ","
header_matrix <- do.call(rbind, header_parts)         # Convert to matrix
header_df <- as.data.frame(header_matrix)             # Convert to data frame
colnames(header_df) <- c("storm_id", "name", "rows") # Name columns

# Repeat each header based on the 'rows' column in tidyr
header_expand <- header_df |>
  mutate(rows = as.numeric(rows)) |>
  uncount(rows)

# Prep data
storm_data <- lines[!grepl("^AL", lines)] # Keep only data
hurdat_parts <- strsplit(storm_data, ",")   # Split based on ","
hurdat_matrix <- do.call(rbind, hurdat_parts) # Convert to matrix
hurdat_df <- as.data.frame(hurdat_matrix)     # Convert to to data frame

hurdat_fields <- c("yyyymmdd", "hhmm", "record", "status",
                   "lat_hemi", "lon_hemi", "wind", "pressure",
                   "ne34", "se34", "sw34", "nw34",
                   "ne50", "se50", "sw50", "nw50",
                   "ne64", "se64", "sw64", "nw64", "radius")

colnames(hurdat_df) <- hurdat_fields # Name columns

# Build analysis ready data set
hurdat_ar <- header_expand |>
  bind_cols(hurdat_df) |> # Glue together storm id and name with data
  mutate(
    yyyymmdd = ymd(yyyymmdd),                    # Tell R this is a date
    hhmm = strptime(hhmm, format = "%H%M"),      # Tell R this is a time
    hhmm = format(hhmm, "%H:%M"),
    lat = as.numeric(substr(lat_hemi, 1, nchar(lat_hemi)-1)), # Remove "S"
    lon = as.numeric(substr(lon_hemi, 1, nchar(lon_hemi)-1)), # Remove "W"
    lat_hemi = substr(lat_hemi, nchar(lat_hemi), nchar(lat_hemi)),
    lon_hemi = substr(lon_hemi, nchar(lon_hemi), nchar(lon_hemi)),
    lat = if_else(lat_hemi == "S", -lat, lat), # Make lat negative if "S"
    lon = if_else(lon_hemi == "W", -lon, lon)  # Make lon negative if "W"
```

```
  ) |>
  mutate_at(c(9:25), as.numeric) |>                  # Make data numeric
  mutate(across(where(is.numeric), ~na_if(., -999))) # Replace NAs

# Save formatted data to read into next quarto environment
saveRDS(hurdat_ar, file = here("assignment2", "hurdat.rds"))
```

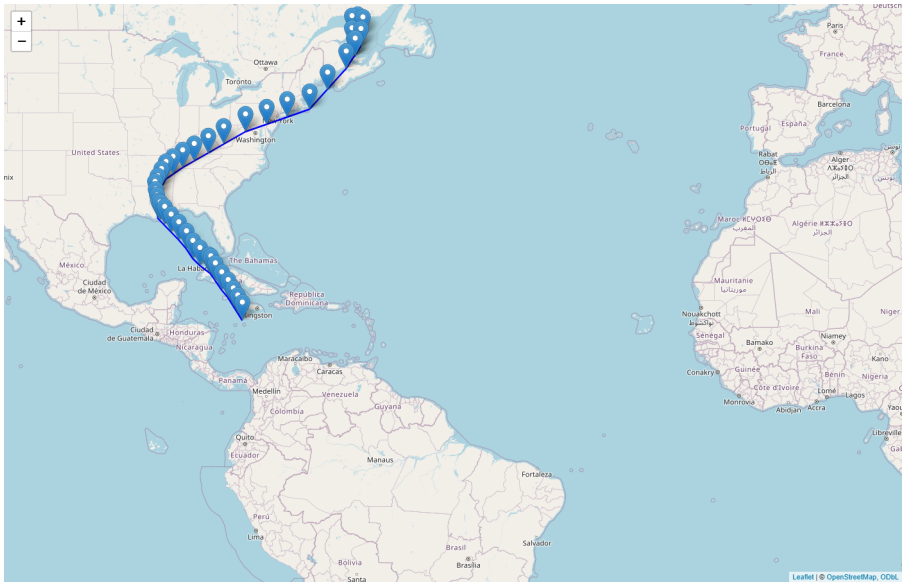
2 Data Visualization

```
# Check if libraries are installed; install if not.
if (!require("pacman")) install.packages("pacman")
pacman::p_load(here, stringr, leaflet, webshot2)
```

```
# Read in data
hurdat_ar <- readRDS("hurdat.Rds")

# Create a reusable function
track_storm <- function(dat, storm_id, zoom = 4,
                            init_location = c(20, -50)) {
  # Filter and order the points for the selected storm
  storm <- dat |>
    filter(storm_id == !!storm_id, !is.na(lat), !is.na(lon)) |>
    mutate(status = str_trim(status)) |>
    arrange(yyyymmdd, hhmm)

  if (nrow(storm) == 0) stop("No points found for this storm_id.")

  # Build popup: date + time + status (e.g., "1851-06-25 00:00 - HU")
  popup_txt <- paste0(
    format(storm$yyyymmdd, "%Y-%m-%d"), " ", storm$hhmm,
    " - ", storm$status
  )

  # Create map
  m <- leaflet(options = leafletOptions(worldCopyJump = TRUE)) |>
    addTiles() |>
    setView(lng = init_location[2], lat = init_location[1], zoom = zoom) |>
    addPolylines(
      lng = storm$lon,
```

```
      lat = storm$lat,
      color = "blue",
      weight = 2.5,
      opacity = 1
    ) |>
    addMarkers(
      lng = storm$lon,
      lat = storm$lat,
      popup = popup_txt
    )

  file_html <- here("assignment2", paste0(storm_id, "_map.html"))
  htmlwidgets::saveWidget(m, file_html, selfcontained = TRUE)
  m
}

leaflet_png <- function(m) {
  file_png = here("assignment2", "hurricane_tracks_map.png")
  html_tmp <- here("assignment2", "hurricane_tracks_map_tmp.html")
  htmlwidgets::saveWidget(m, html_tmp, selfcontained = TRUE)
  webshot2::webshot(html_tmp, file = file_png, vwidth = 1400,
                    vheight = 900, zoom = 1)
  return(file_png)
}
```

```
m <- track_storm(hurdat_ar, storm_id = "AL092021")
png_file <- leaflet_png(m)
knitr::include_graphics(png_file)
```



3 Links to Colab and GitHub

Assignment 2 Google Colab

Quarto book chapter on GitHub