Final Project Markdown

Carson Cherniss & Ainsley Gallagher

Data and Data Cleaning

Dataset:

https://wildlife.faa.gov/home

Research Question:

During what time of day are wildlife collisions most common?

Cleaning in Excel

Dataset over 130MB, too large to upload to GitHub, very slow loading into R. Removed 85 variables to reduce file size: AIRPORT, AIRPORT_LATITUDE, AIRPORT_LONGITUDE, RUNWAY, FAAREGION, LOCATION, OPID, OPERATOR, REG, FLT, AMA, AMO, EMA, EMO, AC_CLASS, AC_MASS, TYPE_ENG, NUM_ENGS, ENG_1_POS, ENG_2_POS, ENG_3_POS, ENG_4_POS, PHASE_OF_FLIGHT, HEIGHT, SPEED, DISTANCE, AOS, COST_REPAIRS, COST_OTHER, COST_REPAIRS_IFL_ADJ, COST_OTHER_IFL_ADJ, INGESTED_OTHER, INDICATED_DAMAGE, DAMAGE_LEVEL, STR_RAD, DAM_RAD, STR_WINDSHLD, DAM_WINDSHLD, STR_NOSE, DAM_NOSE, STR_ENG1, DAM_ENG1, ING_Eng1, STR_ENG2, DAM_ENG2, ING_Eng2, STR_ENG3, DAM_ENG3, ING_Eng3, STR_ENG4, DAM_ENG4, ING_Eng4, STR_PROP, DAM_PROP, STR_WING_ROT, DAM_WING_ROT, STR_FUSE, DAM_FUSE, STR_LG, DAM_LG, STR_TAIL, DAM_TAIL, STR_LIGHTS, DAM_LIGHTS, STR_OTHER, DAM_OTHER, OTHER_SPECIFY, EFFECT, EFFECT_OTHER, BIRD_BAND_NUMBER, OUT_OF_RANGE_SPECIES, REMARKS, REMAINS_COLLECTED, REMAINS_SENT, WARNED, NUM_SEEN, ENROUTE_STATE, NR_INJURIES, NR_FATALATIES, COMMENTS, REPORTED_NAME, REPORTED_TITLE, SOURCE, PERSON, TRANSFER.

Cleaning in R

```r
# Load Packages
pacman::p_load(tidyverse, readxl, lubridate, janitor)
# Read the data
faa_data <- read_excel("Public.xlsx")
glimpse(faa_data)

# Parse INCIDENT_DATE and LUPDATE and TIME as dates
faa_data <- faa_data |>
  mutate(
    INCIDENT_DATE = as_date(INCIDENT_DATE),
    LUPDATE = as_date(LUPDATE),
    TIME = lubridate::hm(TIME)
  )

# Clean names using janitor package
faa_data <- faa_data |>
  janitor::clean_names()
```

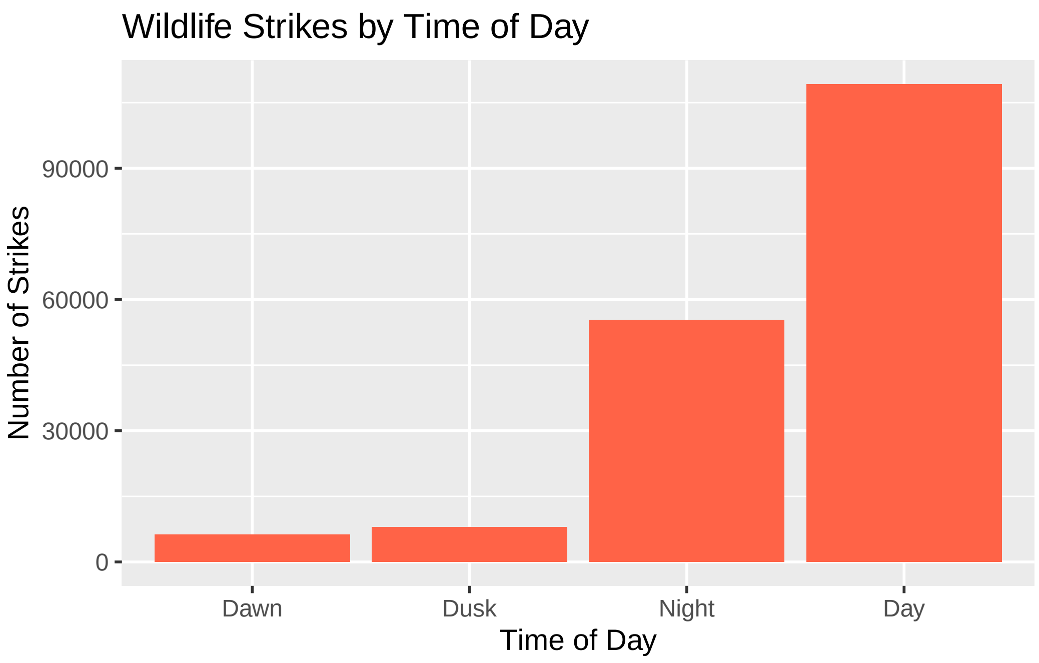
Exploratory Data Analysis

We created data visualizations and conducted a statistical test to investigate our hypothesis.

Data Visualization

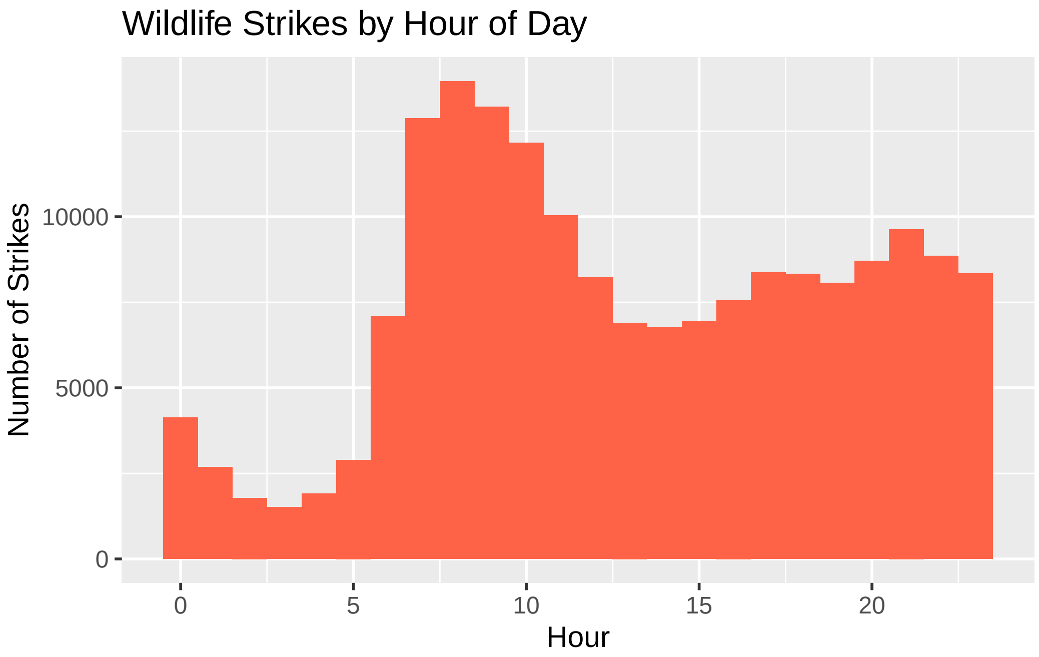
First graph is a bar chart showing how the count of wildlife strikes is distributed by the `time_of_day` variable.

```r
# Graph 1: Bar chart showing time_of_day variable distribution
faa_data |>
  filter(!is.na(time_of_day)) |>
  count(time_of_day) |>
  ggplot(aes(x = fct_reorder(time_of_day, n), y = n)) +
  geom_col(fill = "tomato") +
  labs(
    title = "Wildlife Strikes by Time of Day",
    x = "Time of Day",
    y = "Number of Strikes"
  )
```



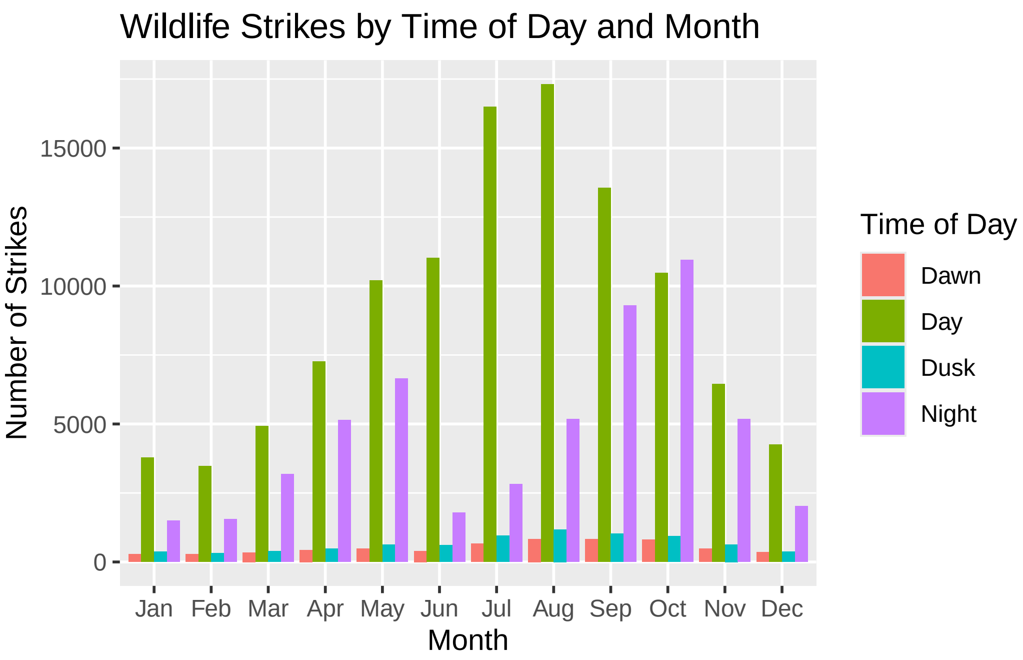
Second graph shows the count but instead uses the `time` variable. This shows more accurately the distribution of strikes throughout the day, since we do not have a codebook for this data and do not know what hours mean `day` or `night` or `dawn` or `dusk`.

```r
# Graph 2: Bar chart showing time variable distribution
faa_data |>
  filter(!is.na(time)) |>
  mutate(hour = hour(time)) |>
  ggplot(aes(x = hour)) +
  geom_histogram(binwidth = 1, fill = "tomato") +
  labs(title = "Wildlife Strikes by Hour of Day",
       x = "Hour",
       y = "Number of Strikes")
```



The final graph will allow us to add an interactive element to the shiny app. The code will change slightly when used in the app, but it will allow us to see the `time_ofday` variable bar graph but broken down by month.

```r
faa_data |>
  filter(!is.na(time_of_day)) |>
  mutate(month = lubridate::month(incident_date, label = TRUE)) |>
  count(month, time_of_day) |>
  ggplot(aes(x = month, y = n, fill = time_of_day)) +
  geom_col(position = "dodge") +
  labs(
    title = "Wildlife Strikes by Time of Day and Month",
    x = "Month",
    y = "Number of Strikes",
    fill = "Time of Day"
  )
```



Statistical Test

Shiny App

The next step was to convert all of this into a shiny app. The code for the app follows.

```
pacman::p_load(tidyverse, readxl, lubridate, janitor, shiny, plotly, shinythemes)

# App setup
ui <- navbarPage(
  title = "FAA Wildlife Strikes",
  theme = shinythemes::shinytheme("readable"),
  tabPanel("Overview",
           fluidPage(
             h3("Project Overview"),
             p("DATA-413 Final Project"),
             p("Carson Cherniss & Ainsley Gallagher"),
             h4("Research Question:"),
             p("During what time of day are wildlife strikes by aircraft most common?"),
             h4("Dataset:"),
             p("The dataset comes from the Federal Aviation Administration (FAA) Wildlife St
             It contains detailed records of wildlife strikes involving civil aircraft in the
             This analysis uses data from 1990 to 2023, with approximately 300,000 observatio
               'incident date,' 'time of day,' 'species,' and 'location.'"),
             h4("Methods:"),
             tags$ul(
               tags$li("Exploratory Data Analysis: Visualizations of wildlife strikes by time
               tags$li("Seasonal Trends: Interactive plot allowing users to view monthly brea
               tags$li("Statistical Test: Chi-square test to evaluate whether wildlife strike
             )
           )
  ),
  tabPanel("Data Visualization: Time of Day",
           fluidPage(
             h3("Wildlife Strikes by Time of Day"),
             plotlyOutput("timeOfDayPlot"),
             p("Based on this visualization, it appears that wildlife strikes are most commo
               We investigate this further with statisitcal testing."),
             p("Without a codebook, we do not know what hours of the day were classified as
               Thus, we can also visualize the distribution of wildlife strikes by hour of th
             h3("Wildlife Strikes by Hour of Day"),
             plotlyOutput("timePlot"),
             p("Here, we can see that strikes appear most common between 7 and 11.
               Wildlife strikes seem common between the hours of 7 and 23. We can investigate
           )
  ),
  tabPanel("Data Visualization: Monthly Trends",
           fluidPage(
             h3("Wildlife Strikes by Month"),
             p("Data visualization has shown that wildlife strikes appear most common during
               Those visualizations are for all strikes throughout the year. However, things
               With this visualization, we can see how trends change throughout the year."),
             selectInput("selected_month", "Select a Month:", choices = month.name),
             plotlyOutput("monthlyPlot"),
             p("Here, we can see that the overall trend of 'day' having the most wildlife st
               However, October has an interesting discrepency, where 'night' has more strik
               This could be due to bird migrations."),
             p("The winter months (January, February and December) as well as the summer mont
               have the greatest difference between the amount of 'day' and 'night' strikes.
               Whereas spring and fall months have a smaller difference between 'day' and 'ni
               as birds tend to migrate in the fall and the spring and may be more active in
           )
  ),
  tabPanel("Statistical Test",
           fluidPage(
             h3("Chi-Square Test Results")
           )
  )
)

# Server
server <- function(input, output) {
  # Load data inside the server so it only loads when the app runs
  faa_data <- reactive({
    read_excel("Public.xlsx") |>
      mutate(
        INCIDENT_DATE = as_date(INCIDENT_DATE),
        LUPDATE = as_date(LUPDATE),
        TIME = lubridate::hm(TIME)
      ) |>
      janitor::clean_names()
  })

  # Bar plot of time_of_day
  output$timeOfDayPlot <- renderPlotly({
    ggplotly(
      faa_data() |>
        filter(!is.na(time_of_day)) |>
        count(time_of_day) |>
        ggplot(aes(x = fct_reorder(time_of_day, n), y = n, fill = time_of_day)) +
        geom_col() +
        labs(title = "Strikes by Time of Day", x = "Time of Day", y = "Number of Strikes")
    )
  })

  # Bar plot of time
  output$timePlot <- renderPlotly({
    ggplotly(
      faa_data() |>
        filter(!is.na(time)) |>
        mutate(hour = hour(time)) |>
        ggplot(aes(x = hour)) +
        geom_histogram(binwidth = 1, fill = "tomato") +
        labs(title = "Wildlife Strikes by Hour of Day",
             x = "Hour",
             y = "Number of Strikes")
    )
  })

  # Monthly plot if we want to keep it
  output$monthlyPlot <- renderPlotly({
    req(input$selected_month)

    ggplotly(
      faa_data() |>
        filter(!is.na(time_of_day)) |>
        mutate(month = lubridate::month(incident_date, label = TRUE, abbr = FALSE)) |>
        filter(month == input$selected_month) |>
        count(time_of_day) |>
        ggplot(aes(x = time_of_day, y = n, fill = time_of_day)) +
        geom_col() +
        labs(
          title = paste("Wildlife Strikes in", input$selected_month),
          x = "Time of Day",
          y = "Number of Strikes",
          fill = "Time of Day"
        )
    )
  })
}

# Run app
shinyApp(ui = ui, server = server)
```