DATASET REPORT

DHANYAPRIYA SOMASUNDARAM

Introduction to UFO Sightings Analysis

In this report, we delve into an intriguing dataset comprising reports of Unidentified Flying Objects (UFOs) sightings across various regions globally. This analysis aims to uncover patterns and trends over time, including the frequency of reports over the years and the commonality of sightings at different times of the day. Such insights may shed light on the behavior of sighting occurrences and their reporting.

1) Time Series Analysis: Illustrates trends over time.

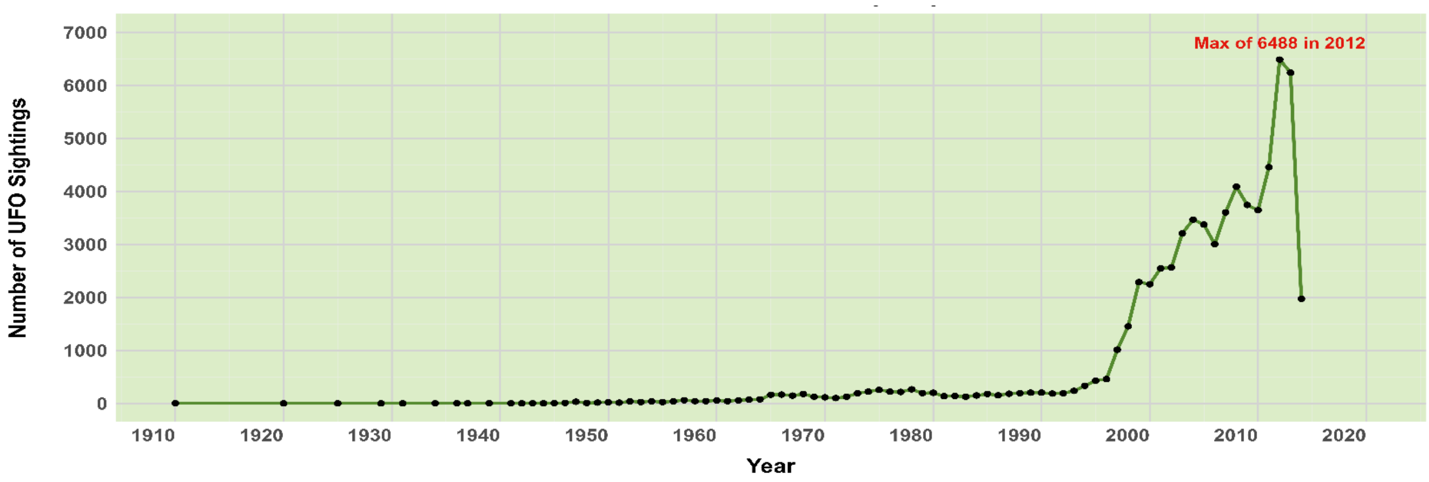


Figure-1: *The line plot above, showcasing decadal trend of UFO Sightings.*

A notable surge is observed as we approach the 21st century, peaking in **2012** with a record high of **6,488 sightings**. This visualization encapsulates the growing public interest and heightened awareness in unidentified aerial phenomena, as evidenced by the data collected.

Distribution of UFO sightings:

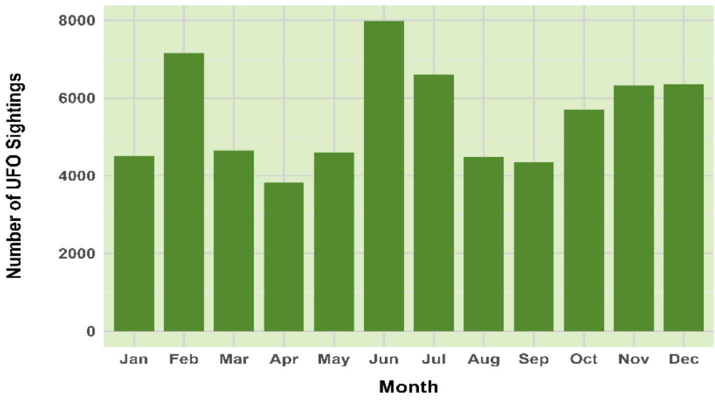


Figure 2.1 depicts the number of UFO sightings reported each month.

Monthly Trend Description:

- The trend shows a clear seasonal pattern, with the lowest number of sightings typically occurring in **April** and a significant increase as the month progresses towards **June**, which has the highest count of reported sightings.
- After June, there is a gradual decrease in the number of reports, though sightings remain relatively high through the end of the year compared to the beginning.

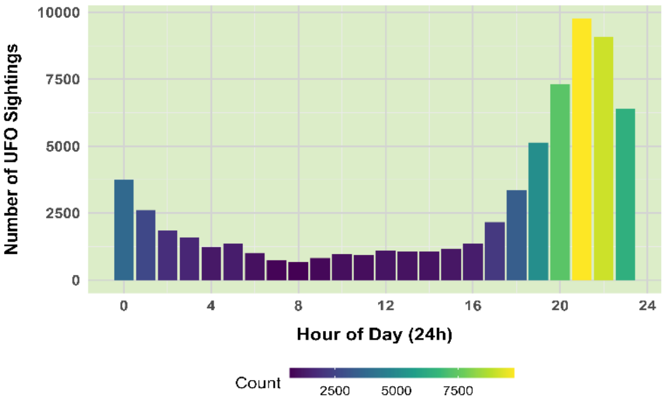


Figure 2.2 Shows the hourly distribution of UFO sightings over a 24-hr period.

Hourly Trend Description:

The number of sightings is relatively low from the **early morning** hours until the afternoon.

- In the **late afternoon**, there is a gradual increase in the number of sightings, with a significant spike starting at around 8 PM, reaching the highest frequency between **9 PM** and midnight.
- **After midnight**, there is a sharp decline in reports, which remains low during the early morning hours.

2) Geographical Analysis:

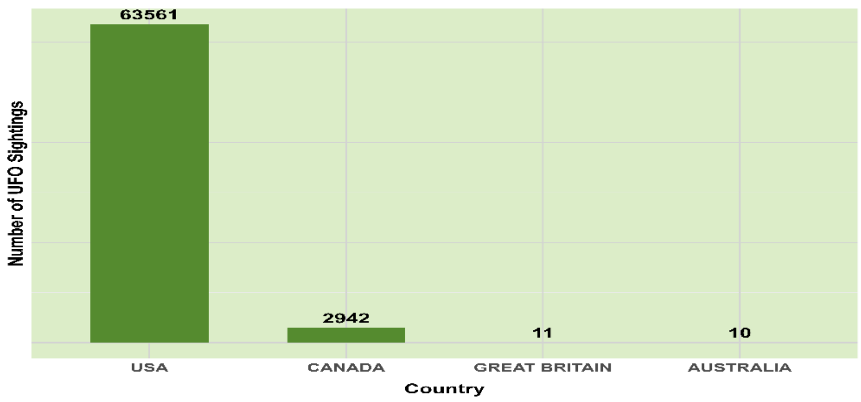


Figure-3.1 *UFO Sightings by Country*

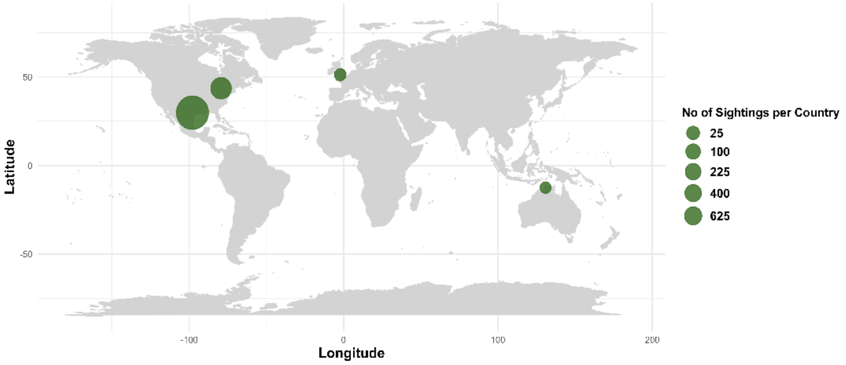


Figure-3.2 *Global Distribution of UFO Sightings*

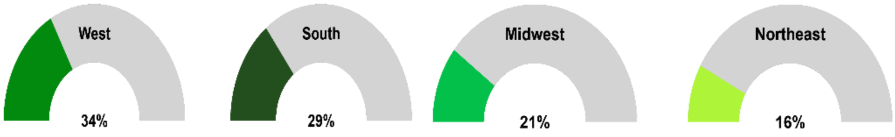


Figure-3.3 *The semi-circle gauge charts, each representing a different region of the United States.*

- *Figure-3.1* Represents bar chart for the Geographical analysis. The trends suggest regional variations in the occurrence or reporting of UFO sightings, with the **USA** leading in the number of reports among the evaluated countries.

- *Figure-3.2* The world map highlights the **USA** as a significant hotspot for UFO reports, with **Canada**, **Great Britain**, and **Australia** presenting considerably fewer incidents by comparison.

- *Figure-3.3* The **West** region leads with the **highest** percentage of UFO sightings 34%, followed by the South at 29%. The Midwest accounts for 21%, while the **Northeast** has the **lowest** at 16%, suggesting a regional trend in sighting occurrences.

3) Descriptive Analysis: The focus is on the distribution of reported Top-10 UFO shapes. The frequency of sightings by shape. Understanding the common shapes observed can offer insights into the nature of these sightings and potentially point to trends in public perception or reporting behavior.

- The shape is **'Light'**, with a significantly **higher** count compared to others.
- There is a clear descending trend in the number of sightings across different shapes, with 'Triangle', 'Circle', and 'Fireball' being the next most frequent shapes reported.
- **Lesser**-reported shapes such as **'Disk'**, **'Oval'**, and **'Formation'** appear at the lower end of the frequency spectrum.

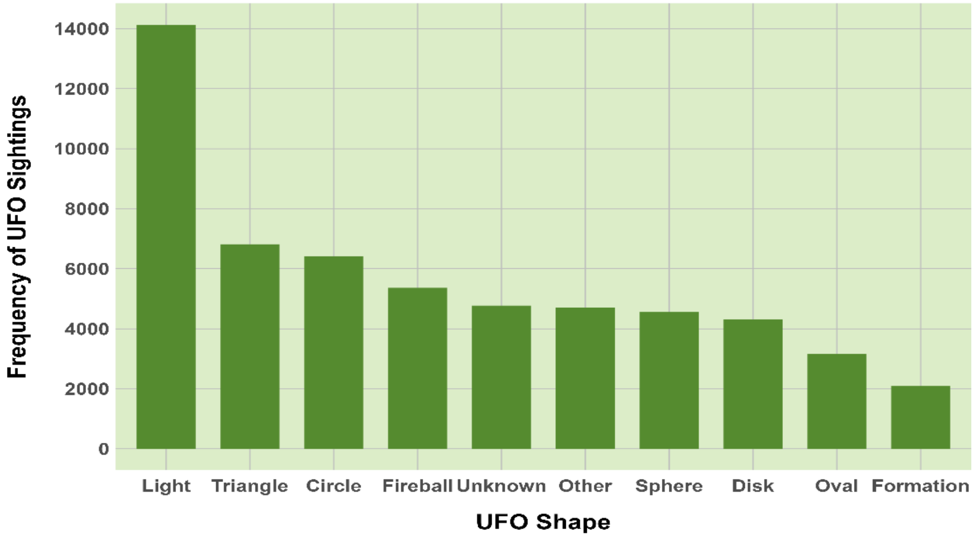
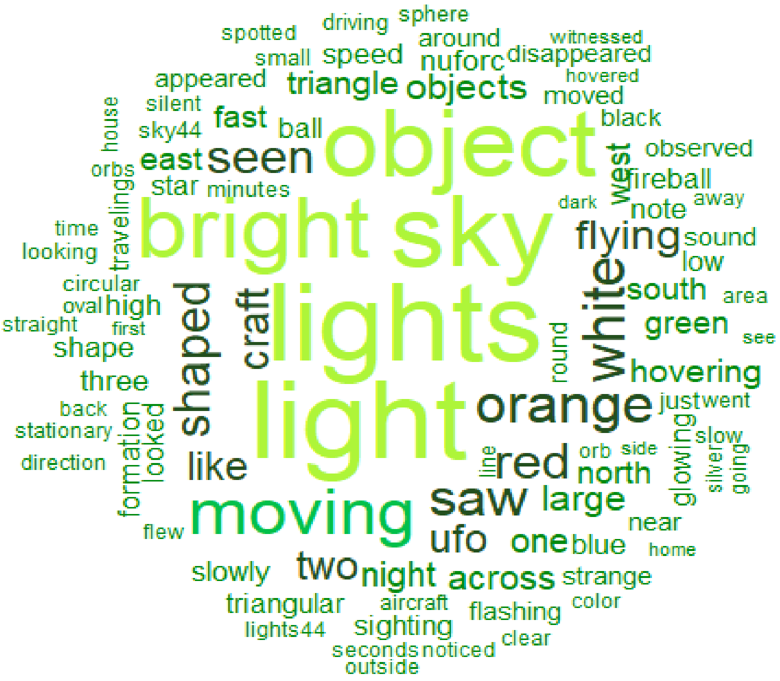


Figure-4: *The bar chart visualizes the top 10 most commonly reported UFO shapes.*

4) Text Analysis: Word clouds

- **Figure-5:** This **word cloud** represents the most frequently mentioned terms in the comments section of the UFO sightings.
- Words like **'lights', 'sky', 'bright', and 'object'** feature prominently, indicating common descriptors used by witnesses.
- The varying sizes of the words reflect their relative frequency, with larger text signifying more common terms in the sightings.



5) Correlation Analysis:

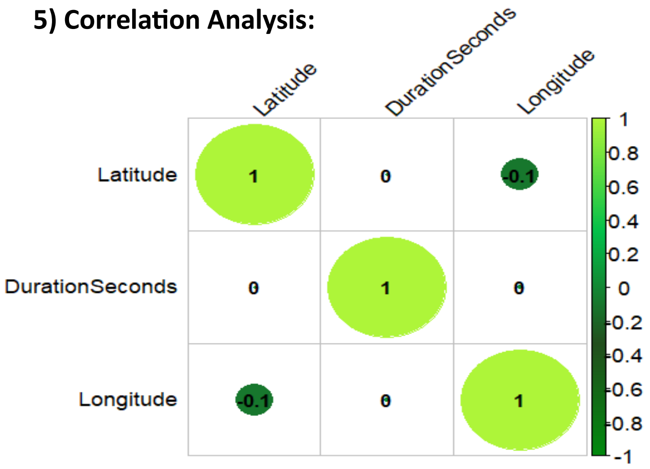


Figure-6 depicts the number of UFO sightings reported each month.

Correlogram (Correlation Matrix Plot): To visualize potential correlations between the duration of sightings and other variables.

Figure-6: This bubble chart depicts the correlation coefficients among 'DurationSeconds', 'Latitude', and 'Longitude' of UFO sightings. **Larger** bubbles represent **stronger** correlations, with **Yellow** indicating **positive** and **Green** indicating **negative** associations. The chart suggests a little to no linear relationship between the geographical coordinates and the duration of sightings.

6) Seasonal Analysis:

- **Figure-7:** The pie chart indicates that UFO sightings are not evenly distributed across seasons, with a noticeable trend showing the **highest** proportion of sightings occurring in the **Fall** (33.2%).
- Summer and Spring follow with 24.8% and 22.1% respectively, suggesting that sightings are more common in warmer months.
- **Winter** has the **lowest** occurrence of sightings at 19.9%, hinting at a possible seasonal influence on the frequency of reported UFO sightings.

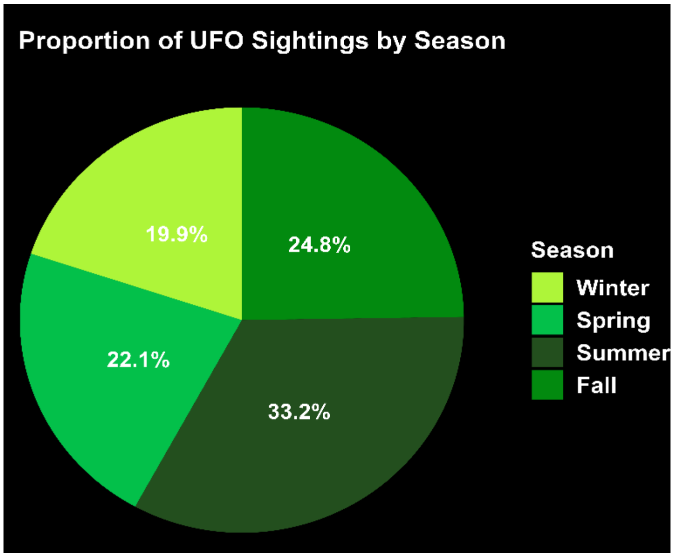


Figure-7

Conclusion:

After analyzing the UFO sightings data, it is evident that certain patterns emerge that may warrant further investigation. Seasonal trends suggest that sightings are more frequent during certain times of the year, possibly linked to weather conditions or human activities. Geographical analysis indicates that sightings are not uniformly distributed, with some regions reporting higher occurrences. The lack of strong correlation between sighting duration and geographical coordinates implies that sightings are independent of location. Overall, this analysis provides a foundational understanding of UFO sighting patterns but suggests the need for deeper analysis to uncover underlying causes or biases in reporting.

Rcode:

```
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(tools)

# Read the dataset
ufo_data <- read.csv("E:/MSDS/SEMESTER-1/INFO-526-DATA VIZ & ANALYSIS-MEGHEAN/Wk-6-Dashboards/ASSIGNMENT-4- Dataset
Report/UFO complete.csv")

# Data Cleaning
# Remove 'date posted' column
ufo_data <- select(ufo_data, -date.posted)

# Rename columns for consistency
colnames(ufo_data) <- c("Datetime", "City", "State", "Country", "Shape",
                        "DurationSeconds", "DurationHoursMin", "Comments",
                        "Latitude", "Longitude")

# Check for missing values and remove rows with missing values
ufo_data <- filter(ufo_data, !is.na(State) & !is.na(City) & !is.na(Country) & !is.na(Shape))
ufo_data <- filter(ufo_data, State != "", Country != "", Shape != "", City != "")

# Convert 'DurationSeconds' to numeric
ufo_data$DurationSeconds <- as.numeric(as.character(ufo_data$DurationSeconds))
ufo_data <- filter(ufo_data, !is.na(DurationSeconds))

# Inspect the data
print(head(ufo_data)) # View the first few rows
print(str(ufo_data)) # Display the structure
print(summary(ufo_data)) # Provide summary statistics

# Check for unique values in the 'Shape' column
unique_shapes <- unique(ufo_data$Shape)
print(unique_shapes)



###################-----------PAGE-1-TIME SERIES ANALYSIS VISUALISATION-----------############
#####################--- 1) Create a line plot for UFO Sightings by Year----########
# Yearly UFO Sightings Trend
ufo_data$Datetime <- as.POSIXct(ufo_data$Datetime, format="%m/%d/%Y %H:%M")  # Convert Datetime column to POSIXct format
ufo_data$Year <- as.integer(format(ufo_data$Datetime, "%Y"))  # Extract the year from the Datetime column

# Group data by year and count the number of sightings per year
year_counts <- ufo_data %>%
  group_by(Year) %>%
  summarise(Count = n())

# Find the row with the maximum Count
max_value <- year_counts[which.max(year_counts$Count), ]

# Create the line plot with points and annotate the year with the maximum sightings
ufo_line_plot <- ggplot(data = year_counts, aes(x = Year, y = Count)) +
  geom_line(color = "#558B2F", group = 1, size = 1) +   # Add a line plot with blue color
  geom_point() +                              # Add points to the plot
  geom_text(data = max_value, aes(label = paste("Max of", Count, "in", Year),fontface = "bold"),
            vjust = -1, hjust = 0.5, color = "#E3170D") +  # Adjust text position
  theme_minimal() +                           # Apply a minimal theme
  labs(x = "Year", y = "Number of UFO Sightings",
       title = "Annual Trend of UFO Sightings")+
  scale_x_continuous(breaks = seq(min(year_counts$Year), 2020, by = 10), limits = c(min(year_counts$Year), 2020)) +  # Set x-axis breaks and limits,
up to 2020 with 10-year intervals
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE), breaks = seq(0,7000, by = 1000), limits = c(0, 7000)) +  # Format y-axis labels
and set breaks at intervals of 1000
  theme(
    panel.background = element_rect(fill = "#DCEDC8", colour = NA), # Set the background color similar to the plot image
    panel.grid.major = element_line(color = "lightgrey", size = 0.5, linetype = "solid"),  # Customize major grid lines
    panel.grid.minor = element_line(color = "#EEEEEE7F", size = 0.25, linetype = "solid"),  # Customize minor grid lines
    axis.text.x = element_text(angle = 0, hjust = 1, face = "bold", size = 12),  # Customize x-axis text and increase font size
    axis.text.y = element_text(face = "bold", size = 12),  # Customize y-axis text and increase font size
    axis.title.y = element_text(face = "bold", margin = margin(r = 20, unit = "pt"), size = 14),  # Customize y-axis title margin and increase font size
    axis.title.x = element_text(face = "bold", margin = margin(t = 10, unit = "pt"), size = 14),  # Customize x-axis title margin and increase font size
    plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title and increase font size
  )
```

```
# Print the UFO line plot
print(ufo_line_plot)



# Save the UFO line plot to a file with adjusted dimensions for a more compact look
graphic_width <- 12  # Adjust the width for a more compact look
graphic_height <- 5  # Adjust the height as needed

# Save the UFO line plot to a PDF file with the adjusted size
ggsave("ufo_line_plot.png", ufo_line_plot, width = graphic_width, height = graphic_height, dpi = 1000)
ggsave("ufo_line_plot.pdf", ufo_line_plot, width = graphic_width, height = graphic_height, dpi = 1000)

##################################################################################################################

############----2) a) Bar Chart by Month-----########################
# Extract the month from the Datetime column
ufo_data$Month <- as.integer(format(ufo_data$Datetime, "%m"))

# Group data by month and count the number of sightings per month
month_counts <- ufo_data %>%
  group_by(Month) %>%
  summarise(Count = n())

# Order the months for plotting (optional, if Month is not already a factor)
month_counts$Month <- factor(month_counts$Month, levels = 1:12, labels = month.name)

# Create the bar chart for sightings by month
ufo_month_bar_chart <- ggplot(data = month_counts, aes(x = substr(Month, 1, 3), y = Count)) +
  geom_bar(stat = "identity", fill = "#558B2F", width=0.8) +   # Use bars to represent counts
  theme_minimal() +                             # Apply a minimal theme
  labs(x = "Month", y = "Number of UFO Sightings") +
  scale_x_discrete(labels = month.abb[match(month_counts$Month, month.name)]) +  # Use three-letter month abbreviations
  theme(
    panel.background = element_rect(fill = "#DCEDC8", colour = NA), # Set the background color similar to the plot image
    panel.grid.major = element_line(color = "lightgrey", size = 0.5, linetype = "solid"),  # Customize major grid lines
    axis.text.x = element_text(angle = 0, hjust = 0.5, face = "bold",size = 10),  # Customize x-axis text and increase font size
    axis.text.y = element_text(face = "bold", size = 10),  # Customize y-axis text and increase font size
    axis.title.y = element_text(face = "bold", margin = margin(r = 20, unit = "pt"), size = 12),  # Customize y-axis title margin and increase font size
    axis.title.x = element_text(face = "bold", margin = margin(t = 10, unit = "pt"), size = 12),  # Customize x-axis title margin and increase font size
  )

# Print the UFO month bar chart
print(ufo_month_bar_chart)


# Save the UFO month bar chart to a file with adjusted dimensions for a more compact look
graphic_width <- 6  # Adjust the width for a more compact look
graphic_height <- 4  # Adjust the height as needed

# Save the UFO month bar chart to a PDF and png file with the adjusted size
ggsave("ufo_month_bar_chartt.png", ufo_month_bar_chart, width = graphic_width, height = graphic_height, dpi = 1000)
ggsave("ufo_month_bar_chart.pdf", ufo_month_bar_chart, width = graphic_width, height = graphic_height, dpi = 1000)


############----2) b) Bar Chart by Hour-----########################

# Extract the hour from the Datetime column
ufo_data$Hour <- as.integer(format(ufo_data$Datetime, "%H"))

# Group data by hour and count the number of sightings per hour
hour_counts <- ufo_data %>%
  group_by(Hour) %>%
  summarise(Count = n())

# Create a sequence of breaks at 4-hour intervals from 0 to 24
hour_breaks <- seq(0, 24, by = 4)

library(scales)  # For additional color scales
```

```
# Create the bar chart for sightings by time of day with gradient colors
ufo_hour_bar_chart <- ggplot(data = hour_counts, aes(x = Hour, y = Count, fill = Count)) +
  geom_bar(stat = "identity") +   # Use bars to represent counts
  scale_fill_gradientn(colors = c("#440154", "#482878", "#3E4A89", "#31688E", "#26828E", "#1F9E89", "#35B779", "#6DCD59", "#B4DE2C", "#FDE725"),
                       space = "Lab",
                       limits = c(min(hour_counts$Count), max(hour_counts$Count)),
                       guide = guide_colorbar(title = "Count", barwidth = 10, barheight = .5)) +
  theme_minimal() +   # Apply a minimal theme
  labs(x = "Hour of Day (24h)", y = "Number of UFO Sightings") +
  scale_x_continuous(breaks = hour_breaks) +  # Set custom breaks on the x-axis
  #scale_y_continuous(breaks = seq(0, 8000, by = 2000), limits = c(0, 8000)) +
  theme(
    panel.background = element_rect(fill = "#DCEDC8", colour = NA), # Set the background color
    panel.grid.major = element_line(color = "lightgrey", size = 0.5, linetype = "solid"),  # Customize major grid lines
    axis.text.x = element_text(angle = 0, hjust = 0.5, face = "bold", size = 10),  # Customize x-axis text
    axis.text.y = element_text(face = "bold", size = 10),  # Customize y-axis text
    axis.title.y = element_text(face = "bold", margin = margin(r = 10, unit = "pt"), size = 12),  # Customize y-axis title margin
    axis.title.x = element_text(face = "bold", margin = margin(t = 10, unit = "pt"), size = 12),  # Customize x-axis title margin
    legend.position = "bottom" # Position the legend at the bottom
  )

# Print the UFO hour bar chart
print(ufo_hour_bar_chart)


# Save the UFO hour bar chart to a file with adjusted dimensions for a more compact look
graphic_width <- 6  # Adjust the width for a more compact look
graphic_height <- 4  # Adjust the height as needed

# Save the UFO hour bar chart to a PDF and png file with the adjusted size
ggsave("ufo_hour_bar_chart.png", ufo_hour_bar_chart, width = graphic_width, height = graphic_height, dpi = 1000)
ggsave("ufo_hour_bar_chart.pdf", ufo_hour_bar_chart, width = graphic_width, height = graphic_height, dpi = 1000)


##########################################################################################################

##################-----------PAGE-2-Geographical Analysis-MAP VISUALISATION-----------###########

# Install and load required libraries
if (!require('ggplot2')) install.packages('ggplot2')
if (!require('ggmap')) install.packages('ggmap')
if (!require('maps')) install.packages('maps')
library(ggplot2)
library(ggmap)
library(maps)


# Ensure 'Latitude' and 'Longitude' are numeric
ufo_data$Latitude <- as.numeric(as.character(ufo_data$Latitude))
ufo_data$Longitude <- as.numeric(as.character(ufo_data$Longitude))

# Group data by country and calculate the number of sightings in each country
country_counts <- ufo_data %>%
  group_by(Country) %>%
  summarise(Sightings = n())  # Count the number of sightings

# Join the country counts with the original UFO data to associate each sighting with its country's count
ufo_data <- left_join(ufo_data, country_counts, by = "Country")

# Create a new variable for bubble size based on the frequency of sightings
ufo_data$BubbleSize <- sqrt(ufo_data$Sightings)  # You can adjust the size transformation as needed

# Group data by country and select the first sighting location within each country
country_data <- ufo_data %>%
  group_by(Country) %>%
  summarise(
    Latitude = first(Latitude),
    Longitude = first(Longitude),
    BubbleSize = first(BubbleSize),
    .groups = "drop"
  )  # Drop grouping
```

```
# Create the map visualization with a light gray fill color and axes titles
ufo_world_map <- ggplot() +
  borders("world", colour =  "#D3D3D3", fill = "#D3D3D3") +  # Light gray color
  geom_point(data = country_data, aes(x = Longitude, y = Latitude, size = BubbleSize), alpha = 0.8, color = "#33691E") +
  theme_minimal() +
  scale_size(name = "No of Sightings per Country", range = c(5, 15), breaks = sqrt(c(25, 100, 225, 400, 625)), labels = c(25, 100, 225, 400, 625)) +  #
Update legend title
  xlab("Longitude") +  # Set x-axis label
  ylab("Latitude") +  # Set y-axis label
  theme(
    axis.title.x = element_text(face = "bold", size = 14),  # Customize x-axis title with bold and size
    axis.title.y = element_text(face = "bold", size = 14),   # Customize y-axis title with bold and size
    legend.text = element_text(size = 12, face="bold"),
    legend.title = element_text(size = 12, face="bold")
  )

# Print the map with updated legend title and bold axes titles
print(ufo_world_map)


# Save the UFO world map to a file with adjusted dimensions for a more compact look
graphic_width <- 12  # Adjust the width for a more compact look
graphic_height <- 5  # Adjust the height as needed

# Save the UFO world map to a PDF and png file with the adjusted size
ggsave("ufo_world_map.png", ufo_world_map, width = graphic_width, height = graphic_height, dpi = 1000)
ggsave("ufo_world_map.pdf", ufo_world_map, width = graphic_width, height = graphic_height, dpi = 1000)


####################################################################################################

##################----PAGE-2 2) Bar Chart by Country or State---########################

# Assuming Country is the column of interest
country_counts <- ufo_data %>%
  group_by(Country) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))

# Bar chart by country with data labels, removed y-axis labels
country_bar_chart <- ggplot(data = country_counts, aes(x = Count, y = reorder(Country, -Count))) +
  geom_bar(stat = "identity", fill = "#558B2F", width = 0.6) +
  geom_text(aes(label = Count), hjust = 0.5, vjust = -0.5, size = 5, fontface = "bold") +  # Add data labels
  theme_minimal() +
  labs(x = "Number of UFO Sightings", y = "Country", title = "UFO Sightings by Country") +
  coord_flip() + # Flip coordinates for horizontal bars
  scale_y_discrete(labels = c("USA", "CANADA", "GREAT BRITAIN", "AUSTRALIA")) +  # Customize x-axis labels
  theme(
    panel.background = element_rect(fill = "#DCEDC8", colour = NA), # Set the background color
    panel.grid.major = element_line(color = "lightgrey", size = 0.5, linetype = "solid"),
    axis.text.x = element_text(face = "bold", size = 12),  # Make x-axis labels bold
    axis.title.x = element_text(face = "bold", margin = margin(t = 10), size = 14),  # Make x-axis title bold
    axis.text.y = element_blank(),  # Remove y-axis labels
    axis.title.y = element_text(face = "bold", margin = margin(r = 0), size = 14),  # Make y-axis title bold
    plot.title = element_text(face = "bold", hjust= 0.5,size = 16),  # Make plot title bold
    plot.caption = element_blank()  # Remove plot caption
  )

# Print the bar chart
print(country_bar_chart)


# Save the country_bar_chart to a file with adjusted dimensions for a more compact look
graphic_width <- 8  # Adjust the width for a more compact look
graphic_height <- 6  # Adjust the height as needed

# Save the country_bar_chart to a PDF and png file with the adjusted size
ggsave("country_bar_chart.png", country_bar_chart, width = graphic_width, height = graphic_height, dpi = 1000)
ggsave("country_bar_chart.pdf", country_bar_chart, width = graphic_width, height = graphic_height, dpi = 1000)

####################################################################################################
```

```
library(dplyr)
library(ggplot2)

# Load your cleaned dataset
#ufo_data <- read.csv("E:/MSDS/SEMESTER-1/INFO-526-DATA VIZ & ANALYSIS-MEGHEAN/Wk-6-Dashboards/ASSIGNMENT-4- Dataset
Report/UFO complete.csv")

# Define the regions as vectors
northeast <- c("CT", "ME", "MA", "NH", "RI", "VT", "NJ", "NY", "PA")
midwest <- c("IL", "IN", "MI", "OH", "WI", "IA", "KS", "MN", "MO", "NE", "ND", "SD")
south <- c("DE", "FL", "GA", "MD", "NC", "SC", "VA", "DC", "WV", "AL", "KY", "MS", "TN", "AR", "LA", "OK", "TX")
west <- c("AZ", "CO", "ID", "MT", "NV", "NM", "UT", "WY", "AK", "CA", "HI", "OR", "WA")

# Ensure State codes are uppercase, as the vectors are in uppercase
ufo_data$State <- toupper(ufo_data$State)

# Categorize states into regions
ufo_data <- ufo_data %>%
  mutate(Region = case_when(
    State %in% northeast ~ "Northeast",
    State %in% midwest ~ "Midwest",
    State %in% south ~ "South",
    State %in% west ~ "West",
    TRUE ~ "Other"
  ))

# Filter data for only US sightings and exclude 'Other' region
ufo_data_us <- filter(ufo_data, Country == "us" & Region != "Other")

# Check if the dataframe is empty after filtering
if(nrow(ufo_data_us) == 0) {
  stop("The filtered dataset has no rows. Check your data and filtering criteria.")
}

# Count the number of sightings per region
region_counts <- ufo_data_us %>%
  count(Region) %>%
  arrange(desc(n))

# Check if the region_counts dataframe is empty
if(nrow(region_counts) == 0) {
  stop("No data available for plotting. Check the region_counts dataframe.")
}

# Ensure that the factor levels for the Region are set in the order you want
region_counts$Region <- factor(region_counts$Region, levels = c("Northeast", "Midwest", "South", "West"))

# Plotting the region-wise bar chart with clear labels
Region_plot <- ggplot(region_counts, aes(x = Region, y = n, fill = Region)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n), vjust = -0.3) +
  labs(title = "UFO Sightings by Region in the US", x = "Region", y = "Number of UFO Sightings") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Print the plot object
print(Region_plot)


# Save the bar chart to a file with adjusted dimensions for a more compact look
graphic_width <- 12  # Adjust the width for a more compact look
graphic_height <- 6  # Adjust the height as needed

# Save the bar chart to a PDF and png file with the adjusted size
ggsave("Region_plot.png", Region_plot, width = graphic_width, height = graphic_height, dpi = 1000)
ggsave("Region_plot.pdf", Region_plot, width = graphic_width, height = graphic_height, dpi = 1000)
####################################################################################################
```

```
library(dplyr)
library(ggplot2)

print(head(ufo_data_us))

ufo_region <- ufo_data_us %>%
  filter(Country == "us", State != "unknown") %>%
  group_by(Region) %>%
  summarize(Count = n(), .groups = 'drop') # Ensure groups are dropped after summarization

ufo_gauge <- ufo_region %>%
  mutate(Percentage = Count / sum(Count))

df <- data.frame(
  variable = ufo_gauge$Region,
  percentage = ufo_gauge$Percentage,
  title = rep(unique(ufo_gauge$Region), nrow(ufo_gauge))
)

df <- df %>%
  arrange(desc(percentage)) %>%
  mutate(variable = factor(variable, levels = unique(variable)))  # Order the factor levels based on the percentage

# Your custom colors
custom_colors <- c("#028A0F", "#234F1E", "#03C04A", "#AEF539")

# Modify  plot code to use the custom colors
plot_regions <- ggplot(df, aes(fill = variable, ymax = percentage, ymin = 0, xmax = 2, xmin = 1)) +
  geom_rect(aes(ymax = 1, ymin = 0, xmax = 2, xmin = 1), fill = "lightgray") +
  geom_rect() +
  coord_polar(theta = "y", start = -pi / 2) + xlim(c(0, 2)) + ylim(c(0, 2)) +
  geom_text(aes(x = 0, y = 0, label = paste0(round(percentage * 100), "%")), size = 1.5, color = "black", fontface="bold") +
  geom_text(aes(x = 1.5, y = 0.5, label = title), size = 1.5, fontface="bold") +
  scale_fill_manual(values = custom_colors) +  # Set custom fill colors
  facet_wrap(~variable, ncol = 2) +
  theme_void() +
  theme(strip.background = element_blank(),
        strip.text.x = element_blank(),
        panel.spacing = unit(0, "lines")) +
  guides(fill = FALSE, color = FALSE)


print(plot_regions)

# save the plot_regions as an image file
ggsave("plot_regions.png", plot_regions, width = 4, height = 2, dpi = 1000)

# Save the plot_regions as a PDF file
ggsave("plot_regions.pdf", plot_regions, width = 4, height = 2 , dpi = 1000)




##############################################################################################################################
################## BAR CHART==The top 10 most common shapes-----################################
ufo_shapes <- table(ufo_data$Shape)
ufo_shapes_df <- as.data.frame(ufo_shapes)
# Convert 'Shape' to character and then apply toTitleCase
ufo_shapes_df$Var1 <- as.character(ufo_shapes_df$Var1)
ufo_shapes_df$Var1 <- toTitleCase(ufo_shapes_df$Var1)

# Convert back to factor if needed for the plot
ufo_shapes_df$Var1 <- as.factor(ufo_shapes_df$Var1)

# Modify the bar chart to show only the top 10 most common shapes and set y-axis breaks
ufo_shapes_sorted <- ufo_shapes_df %>%
  arrange(desc(Freq)) %>%
  top_n(10, Freq)
```

```
# Create the bar chart for the top 10 shapes with specified axis modifications and descending order
top_shapes_bar_chart <- ggplot(ufo_shapes_sorted, aes(x = reorder(Var1, -Freq), y = Freq)) +
  geom_bar(stat = "identity", fill = "#558B2F", width = 0.7) +
  scale_y_continuous(breaks = seq(0, max(ufo_shapes_sorted$Freq), by = 2000)) +
  theme_minimal() +
  labs(x = "UFO Shape", y = "Frequency of UFO Sightings") +
  theme(
        panel.background = element_rect(fill = "#DCEDC8", colour = NA), # Set the background color
        panel.grid.major = element_line(color = "gray", size = 0.2), # Customize gridlines
        panel.grid.minor = element_blank(), # Remove minor gridlines
        axis.text.y = element_text(face = "bold", margin = margin(r = 0)),
        axis.text.x = element_text(face = "bold"),
        axis.title.x = element_text(face = "bold", margin = margin(t = 10)),
        axis.title.y = element_text(face = "bold", margin = margin(r = 10)),
        plot.title = element_text(hjust = 0.5), # Center the title
        plot.caption = element_text(hjust = 0))  # Left-align the plot caption

# Print the modified top shapes bar chart with increased visibility of gridlines
print(top_shapes_bar_chart)

# save the top shapes bar chart as an image file
ggsave("top_shapes_bar_chart.png", top_shapes_bar_chart, width = 6, height = 4, dpi = 1000)

# Save the modified top shapes bar chart as a PDF file
ggsave("top_shapes_bar_chart.pdf", top_shapes_bar_chart, width = 6, height = 4 , dpi = 1000)



###########################################################################################

##############-------Page-3-Text Analysis: Word Cloud------------######################

# Install and load the required library for text analysis
if (!require('tm')) install.packages('tm')
if (!require('wordcloud')) install.packages('wordcloud')
library(tm)
library(wordcloud)

# Prepare the text data
comments_corpus <- Corpus(VectorSource(ufo_data$Comments))
comments_corpus <- tm_map(comments_corpus, content_transformer(tolower))
comments_corpus <- tm_map(comments_corpus, removePunctuation)
comments_corpus <- tm_map(comments_corpus, removeWords, stopwords("english"))
comments_corpus <- tm_map(comments_corpus, stripWhitespace)

# Generate the word cloud
set.seed(1234) # for reproducibility
ufo_word_cloud <- wordcloud(comments_corpus, max.words = 100, random.order = FALSE, colors = c("#028A0F", "#234F1E", "#03C04A", "#AEF539")
)

# Save the word cloud to a file
# Print the modified top shapes bar chart with increased visibility of gridlines
print(ufo_word_cloud)

# Save the word cloud as a PNG file
png("ufo_word_cloud.png", width = 800, height = 800, res = 1000)
dev.off()

#################################################################################################
#############----Correlation Analysis:bubble heat map-----##############################
# Install and load the corrplot package for correlation matrix visualization
if (!require('corrplot')) install.packages('corrplot')
library(corrplot)

# Calculate the correlation matrix
numeric_data <- ufo_data %>%
  select(DurationSeconds, Latitude, Longitude) %>%
  na.omit()  # Remove NAs to calculate correlations
cor_matrix <- cor(numeric_data)
```

```
# Load the corrplot library
library(corrplot)

# Define the color palette based on the provided color codes
color_palette <- colorRampPalette(c("#028A0F", "#234F1E", "#03C04A", "#AEF539"))

# Create the correlogram using the custom color palette
correlogram_custom_color<-corrplot(cor_matrix, method = "circle", col = color_palette(200),
         type = "full", order = "hclust",
         addCoef.col = "black", # Add coefficient color
         tl.col = "black", # Text label color
         tl.srt = 45, # Text label rotation
         tl.cex = 1.1, # Text label size for the correlation matrix
         cl.cex = 1.1, # Color legend text size
         cl.ratio = 0.2, # Color legend size
         mar = c(1,1,2,1)) # Margins around the plotc(bottom, left, top, right)
#tl.pos = "lt" # Text label position (left and top sides of the cell)
#tl.offset = 0.5 # Text label offset (the amount of space between the cell and the labels)



# Save the correlogram to a file
png("correlogram_custom_colors.png", width = 500, height = 500, res = 1000)
dev.off()

############----Season PIE Chart-------############################

# Convert 'Datetime' to Date object if it's not already
ufo_data$Datetime <- as.Date(ufo_data$Datetime, format="%m/%d/%Y %H:%M")

# Function to assign a season to a date
get_season <- function(date) {
  year <- format(date, "%Y")
  spring_start <- as.Date(paste(year, "-03-21", sep=""))
  summer_start <- as.Date(paste(year, "-06-21", sep=""))
  fall_start <- as.Date(paste(year, "-09-23", sep=""))
  winter_start <- as.Date(paste(year, "-12-21", sep=""))

  if (date >= spring_start & date < summer_start) {
    return("Spring")
  } else if (date >= summer_start & date < fall_start) {
    return("Summer")
  } else if (date >= fall_start & date < winter_start) {
    return("Fall")
  } else {
    return("Winter")
  }
}

# Apply the function to add a 'Season' column
ufo_data$Season <- sapply(ufo_data$Datetime, get_season)

# Aggregate data by Season
season_counts <- ufo_data %>%
  group_by(Season) %>%
  summarise(Count = n()) %>%
  mutate(Season = factor(Season, levels = c("Winter", "Spring", "Summer", "Fall"))) # Order the levels of Season

# Convert counts to a proportion
season_counts$Proportion <- season_counts$Count / sum(season_counts$Count)
```

```
# Create the pie chart for sightings per season
season_pie_chart <- ggplot(season_counts, aes(x = "", y = Proportion, fill = Season)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  scale_fill_manual(values = c("Fall" = "#028A0F", "Summer" = "#234F1E", "Spring" = "#03C04A", "Winter" = "#AEF539")) +
  theme_void() +
  labs(fill = "Season", title = "Proportion of UFO Sightings by Season") +
  theme(legend.position = "right",
        legend.text = element_text(size = 12, face="bold"),
        legend.title = element_text(size = 12, face="bold"),
        plot.title = element_text(hjust = 0.8, color = "white", face="bold"),  # Center the title and set text color to white
        plot.background = element_rect(fill = "black"),  # Set the background color to black
        panel.background = element_rect(fill = "black"),  # Set the panel background color to black
        panel.grid = element_blank(),  # Remove grid lines
        axis.text = element_blank(),  # Remove axis text
        axis.title = element_blank(),  # Remove axis titles,
        text = element_text(color = "white")) +
  geom_text(aes(label = paste0(round(Proportion * 100, 1), "%"), fontface = "bold"),
            position = position_stack(vjust = 0.5), size = 4, color = "white")

# Print the pie chart with bold labels
print(season_pie_chart)


# Save the country_bar_chart to a file with adjusted dimensions for a more compact look
graphic_width <- 5  # Adjust the width for a more compact look
graphic_height <- 5  # Adjust the height as needed

# Save the pie chart as an image file
ggsave("season_pie_chart.png", season_pie_chart, width = graphic_width, height =graphic_height, dpi = 1000)

# Save the pie chart as a PDF file
ggsave("season_pie_chart.pdf", season_pie_chart, width = graphic_width, height = graphic_height, dpi = 1000)
```