quadrillionaiire /
Phase-1-Project

Code Issues Pull requests 1 Actions Projects 1 Wiki Secu

main Phase-1-Project / notebooks / clean_notebook.ipynb

quadrillionaiire finalize notebook 10b60a3 · 4 minutes ago

1 lines (1 loc) · 1.23 MB

National UFO Reporting Center Data Analysis

Overview

This project analyzes UFO sighting data collected by the National UFO Reporting Center (NUFORC), which has recorded over 80,000 reports from 1949 to 2013. By conducting a descriptive analysis of sighting locations, timings, and characteristics, we aim to uncover patterns and trends in UFO sightings. Our analysis explores key questions such as the most commonly reported UFO shapes, the times and places where sightings are most frequent, and whether any correlations exist between the time of sighting and the likelihood of a UFO encounter.

Business Problem

By identifying patterns in UFO sightings, the National UFO Reporting Center can enhance its ability to communicate findings to researchers, enthusiasts, and governmental entities. Providing clear, data-driven visualizations of UFO sighting frequencies, shapes, and encounter characteristics will improve the understanding of these phenomena for tourist, government, or aerospace agencies. This, in turn, will support strategic decisions related to future research, public education, and potential resource allocation for investigating UFO encounters.

Data Understanding

The dataset contains detailed information on UFO sightings reported globally. It includes key features like the date and time of the sighting, geographic information (country, region, and locale), descriptions of the UFO (shape and encounter duration), and other attributes that can be used for exploratory data analysis and predictive modeling. Given the diverse nature of this dataset, we will focus on cleaning and processing the data to answer three key business questions:

1. What regions and times have the highest frequency of UFO sightings?
   - We will use the Country, Region, Locale, Year, Month, Hour, and Season columns to find the hotspots for UFO activity.
2. Are there notable patterns in UFO shapes, descriptions, or lengths of encounters?
   - The columns UFO_shape, length_of_encounter_seconds, and Description will be analyzed to explore common shapes, durations, and narratives in the reported encounters.
3. Can any correlations be drawn between the timing (season, time of day) and the likelihood of a sighting? (Is there a potential for identifying anomalies or "false positives" in the sighting reports?)
   - We will explore correlations between the Season, Month, Hour, and length_of_encounter_seconds to see if UFO sightings show patterns based on the time of day or season of the year.

In [48]:

```
#Import and load the data with pandas

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import matplotlib.colors as mcolors

import os
os.chdir("/Users/saniaspry/Documents/Flatiron/Assignments/Phase1/Phase-1 I

%matplotlib inline
```

In [49]:

```
ufo_df = pd.read_csv("data/ufo_data/ufo-sightings-transformed.csv", index_
```

In [50]:

```
ufo_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 80328 entries, 0 to 80327
Data columns (total 16 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Date_time                      80328 non-null  object
 1   date_documented                80328 non-null  object
 2   Year                           80328 non-null  int64
 3   Month                          80328 non-null  int64
 4   Hour                           80328 non-null  int64
 5   Season                         80328 non-null  object
 6   Country_Code                   80069 non-null  object
 7   Country                        80069 non-null  object
 8   Region                         79762 non-null  object
 9   Locale                         79871 non-null  object
 10  latitude                       80328 non-null  float64
 11  longitude                      80328 non-null  float64
 12  UFO_shape                      78398 non-null  object
 13  length_of_encounter_seconds    80328 non-null  float64
 14  Encounter_Duration             80328 non-null  object
 15  Description                    80313 non-null  object
dtypes: float64(3), int64(3), object(10)
memory usage: 10.4+ MB
```

UFO Data

The UFO sighting datset includes over 80,000 records of UFO sightings from October 1949 to September 2013, and captures a wide variety of sighting locations, times, and characteristics, such as UFO shapes and encounter durations.

In [51]:

```
ufo_df.head()
```

Out[51]:

	Date_time	date_documented	Year	Month	Hour	Season	Country_Code	Count
0	1949-10-10 20:30:00	4/27/2004	1949	10	20	Autumn	USA	Unit Stat
1	1949-10-10 21:00:00	12/16/2005	1949	10	21	Autumn	USA	Unit Stat

2	1955-10-10 17:00:00	1/21/2008	1955	10	17	Autumn	GBR	Unit Kingdc
3	1956-10-10 21:00:00	1/17/2004	1956	10	21	Autumn	USA	Unit Stat
4	1960-10-10 20:00:00	1/22/2004	1960	10	20	Autumn	USA	Unit Stat

In [52]:

```
ufo_df['Sighting Date'] = pd.to_datetime(ufo_df['Date_time'])
ufo_df['Sighting Date'].describe()
```

Out[52]:

```
count                          80328
mean     2004-05-17 07:19:24.235882880
min                1906-11-11 00:00:00
25%                2001-08-02 22:25:00
50%                2006-11-22 05:57:00
75%                2011-06-21 03:30:00
max                2014-05-08 18:45:00
Name: Sighting Date, dtype: object
```

In [53]:

```
ufo_df['Locale'].describe()
```

Out[53]:

```
count           79871
unique          13245
top       Los Angeles
freq              827
Name: Locale, dtype: object
```

In [54]:

```
ufo_df['UFO_shape'].describe()
```

Out[54]:

```
count     78398
unique       29
top       Light
freq      16565
Name: UFO_shape, dtype: object
```

In [55]:

```
ufo_df['length_of_encounter_seconds'].describe()
```

main Phase-1-Project / notebooks / clean_notebook.ipynb Top

Preview Code Blame Raw

```
50%      1.800000e+02
75%      6.000000e+02
max      9.783600e+07
```

```
Name: length_of_encounter_seconds, dtype: float64
```

In [56]:
```
ufo_df['Hour'].describe()
```

Out[56]:
```
count    80328.000000
mean        15.525172
std          7.753750
min          0.000000
25%         10.000000
50%         19.000000
75%         21.000000
max         23.000000
Name: Hour, dtype: float64
```

Data Cleaning

In [57]:
```
#Check for missing values
print(ufo_df.isnull().sum())
```

```
Date_time                         0
date_documented                   0
Year                              0
Month                             0
Hour                              0
Season                            0
Country_Code                    259
Country                         259
Region                          566
Locale                          457
latitude                          0
longitude                         0
UFO_shape                      1930
length_of_encounter_seconds       0
Encounter_Duration                0
Description                      15
Sighting Date                     0
dtype: int64
```

Handling Missing Data

After reading and getting a general overview of our data, we can see missing values in the country code, country, region, locale, UFO shape, and description columns to clean.

In [58]:
```
#Decide to just drop the rows with missing values in the Country column
# Drop rows with missing values in 'Country_Code', 'Country', 'Region', 'L
ufo_df.dropna(subset=['Country_Code', 'Country', 'Region', 'Locale'], inp
print(ufo_df)
```

```
                Date_time date_documented  Year  Month  Hour  Season  \
0     1949-10-10 20:30:00       4/27/2004  1949     10    20  Autumn
1     1949-10-10 21:00:00      12/16/2005  1949     10    21  Autumn
2     1955-10-10 17:00:00       1/21/2008  1955     10    17  Autumn
3     1956-10-10 21:00:00       1/17/2004  1956     10    21  Autumn
4     1960-10-10 20:00:00       1/22/2004  1960     10    20  Autumn
```

```
...                    ...              ...   ...    ...   ...     ...
80323  2013-09-09 21:15:00       9/30/2013  2013      9    21  Autumn
80324  2013-09-09 22:00:00       9/30/2013  2013      9    22  Autumn
80325  2013-09-09 22:00:00       9/30/2013  2013      9    22  Autumn
80326  2013-09-09 22:20:00       9/30/2013  2013      9    22  Autumn
80327  2013-09-09 23:00:00       9/30/2013  2013      9    23  Autumn

      Country_Code         Country      Region        Locale   latitude  \
0              USA   United States       Texas    San Marcos  29.883056
1              USA   United States       Texas  Bexar County  29.384210
2              GBR  United Kingdom     England       Chester  53.200000
3              USA   United States       Texas          Edna  28.978333
4              USA   United States      Hawaii       Kaneohe  21.418056
...            ...             ...         ...           ...        ...
80323          USA   United States   Tennessee     Nashville  36.165833
80324          USA   United States       Idaho         Boise  43.613611
80325          USA   United States  California    Napa Abajo  38.297222
80326          USA   United States    Virginia        Vienna  38.901111
80327          USA   United States    Oklahoma        Edmond  35.652778

        longitude UFO_shape  length_of_encounter_seconds Encounter_Duration
\
0      -97.941111  Cylinder                       2700.0         45 minutes
1      -98.581082     Light                       7200.0            1-2 hrs
2       -2.916667    Circle                         20.0         20 seconds
3      -96.645833    Circle                         20.0           1/2 hour
4     -157.803611     Light                        900.0         15 minutes
...           ...       ...                          ...                ...
80323  -86.784444     Light                        600.0         10 minutes
80324 -116.202500    Circle                       1200.0         20 minutes
80325 -122.284444     Other                       1200.0               hour
80326  -77.265556    Circle                          5.0          5 seconds
80327  -97.477778     Cigar                       1020.0         17 minutes

                                             Description       Sighting Dat
e
0      This event took place in early fall around 194... 1949-10-10 20:30:0
0
1      1949 Lackland AFB&#44 TX.  Lights racing acros... 1949-10-10 21:00:0
0
2      Green/Orange circular disc over Chester&#44 En... 1955-10-10 17:00:0
0
3      My older brother and twin sister were leaving ... 1956-10-10 21:00:0
0
4      AS a Marine 1st Lt. flying an FJ4B fighter/att... 1960-10-10 20:00:0
0
...                                                  ...
...
80323  Round from the distance/slowly changing colors... 2013-09-09 21:15:0
0
80324  Boise&#44 ID&#44 spherical&#44 20 min&#44 10 r... 2013-09-09 22:00:0
0
80325                                      Napa UFO&#44 2013-09-09 22:00:0
0
80326  Saw a five gold lit cicular craft moving fastl... 2013-09-09 22:20:0
0
80327  2 witnesses 2  miles apart&#44 Red &amp; White... 2013-09-09 23:00:0
0

[79588 rows x 17 columns]
```

At first, we tried using the geopandas, geopy, and geocoder to convert latitude and longitude coordinates to countries. However, this did not work, resulting in a updated CSV file with missing values, so we decided to drop the rows with missing values in the country_code, country, region, and locale column because we realized that most of our data might have taken place over international water

For the UFO shape, since there is already an unknown column we are going to fill in our missing values with unknown. We will also fill in the missing descriptions with a placeholder.

In [59]:

```
#fill missing UFO shapes with a placeholder
ufo_df['UFO_shape'].fillna('Unknown', inplace=True)

# fill descriptions with a placeholder or leave them as is
ufo_df['Description'].fillna('No description provided', inplace=True)
```

```
/var/folders/p9/l56kxrqj1f50k63kvkm8k0nm0000gp/T/ipykernel_19406/143722118
8.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame
or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behav
es as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) in
stead, to perform the operation inplace on the original object.


  ufo_df['UFO_shape'].fillna('Unknown', inplace=True)
/var/folders/p9/l56kxrqj1f50k63kvkm8k0nm0000gp/T/ipykernel_19406/143722118
8.py:5: FutureWarning: A value is trying to be set on a copy of a DataFrame
or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behav
es as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) in
stead, to perform the operation inplace on the original object.


  ufo_df['Description'].fillna('No description provided', inplace=True)
```

In [60]:

```
# Verify changes to ensure rows with missing values in the specified colu
print(ufo_df[['Country_Code', 'Country', 'Region', 'Locale', 'UFO_shape']
```

```
Country_Code    0
Country         0
Region          0
Locale          0
UFO_shape       0
dtype: int64
```

Additional Standarizations

Since the date time, encouter duration, date documentated, year, month, hour, column time units vary we are going to convert these columns into a consistent format for analysis using the pd.to_datetime() and numeric().

As an example lets take a look at a few sample values in the Encounter Duration column

In [61]:
```
# Display unique values to understand various formats in 'Encounter_Durat
ufo_df['Encounter_Duration'].unique()[:20]  # Display first 20 unique valu
```

Out[61]:
```
array(['45 minutes', '1-2 hrs', '20 seconds', '1/2 hour', '15 minutes',
       '5 minutes', 'about 3 mins', '20 minutes', '3  minutes',
       'several minutes', '5 min.', '3 minutes', '30 min.', '30 seconds',
       '20minutes', '2 minutes', '20-30 min', '20 sec.', 'one hour?',
       '5-6 minutes'], dtype=object)
```

Time Standardization

As we can see the units have already been unified in the length of encounter in seconds column

In [62]:
```
#Standardize Dates and Times
#Convert Date_time to datetime format
ufo_df['Date_time'] = pd.to_datetime(ufo_df['Date_time'], errors='coerce'
ufo_df.head()
```

Out[62]:

	Date_time	date_documented	Year	Month	Hour	Season	Country_Code	Count
0	1949-10-10 20:30:00	4/27/2004	1949	10	20	Autumn	USA	Unit Stat
1	1949-10-10 21:00:00	12/16/2005	1949	10	21	Autumn	USA	Unit Stat
2	1955-10-10 17:00:00	1/21/2008	1955	10	17	Autumn	GBR	Unit Kingdc
3	1956-10-10 21:00:00	1/17/2004	1956	10	21	Autumn	USA	Unit Stat
4	1960-10-10 20:00:00	1/22/2004	1960	10	20	Autumn	USA	Unit Stat

In [63]:

```
#Standardize the date_documented column
ufo_df['date_documented'] = pd.to_datetime(ufo_df['date_documented'], erro
ufo_df.head()
```

Out[63]:

	Date_time	date_documented	Year	Month	Hour	Season	Country_Code	Count
0	1949-10-10 20:30:00	2004-04-27	1949	10	20	Autumn	USA	Unit Stat
1	1949-10-10 21:00:00	2005-12-16	1949	10	21	Autumn	USA	Unit Stat
2	1955-10-10 17:00:00	2008-01-21	1955	10	17	Autumn	GBR	Unit Kingdc
3	1956-10-10 21:00:00	2004-01-17	1956	10	21	Autumn	USA	Unit Stat
4	1960-10-10 20:00:00	2004-01-22	1960	10	20	Autumn	USA	Unit Stat

In [64]:

```
#Standardize Year, Month, Hour and Season Column
#Year, Month, and Hour : are they derived accurately from Date_time?
#Season: grouping months into seasons
ufo_df['Year'] = ufo_df['Date_time'].dt.year
ufo_df['Month'] = ufo_df['Date_time'].dt.month
ufo_df['Hour'] = ufo_df['Date_time'].dt.hour

#Seasons Standardization
ufo_df['Season'] = ufo_df['Month'].apply(lambda x: 'Winter' if x in [12, 1

ufo_df.head()
```

Out[64]:

	Date_time	date_documented	Year	Month	Hour	Season	Country_Code	Count
0	1949-10-10 20:30:00	2004-04-27	1949	10	20	Autumn	USA	Unit Stat
1	1949-10-							Unit

1	10 21:00:00	2005-12-16	1949	10	21	Autumn	USA	Stat
2	1955-10-10 17:00:00	2008-01-21	1955	10	17	Autumn	GBR	Unit Kingdc
3	1956-10-10 21:00:00	2004-01-17	1956	10	21	Autumn	USA	Unit Stat
4	1960-10-10 20:00:00	2004-01-22	1960	10	20	Autumn	USA	Unit Stat

In [65]:

```
#Standardize the length of encounter seconds column
# Convert 'length_of_encounter_seconds' to numeric, forcing errors to NaN
ufo_df['length_of_encounter_seconds'] = pd.to_numeric(ufo_df['length_of_er

# Remove outliers (e.g., encounters longer than a day)
ufo_df = ufo_df[ufo_df['length_of_encounter_seconds'] <= 86400]  # 86400 s

# Verify changes
ufo_df['length_of_encounter_seconds'].describe()
```

Out[65]:

```
count    79413.000000
mean       927.032450
std       3422.613148
min          0.001000
25%         30.000000
50%        180.000000
75%        600.000000
max      86400.000000
Name: length_of_encounter_seconds, dtype: float64
```

Handling Geographical Data

We are going to standardize country codes and names as well to handle any inconsistent entries such as case sensity or misspellings

In [66]:

```
#Standardize Country Codes and Names.
#Ensure that the country codes and Country values are consistent.
ufo_df['Country_Code']= ufo_df['Country_Code'].str.upper().str.strip()
ufo_df['Country'] = ufo_df['Country'].str.title().str.strip()

ufo_df['Country_Code'].unique()
```

Out[66]:

```
array(['USA', 'GBR', 'BMU', 'CAN', 'NZL', 'RUS', 'AUS', 'LTU', 'NOR',
       'ISL', 'MEX', 'AUT', 'VNM', 'BEL', 'CHN', 'GRC', 'FRA', 'CHL',
       'IDN', 'IND', 'THA', 'ESP', 'MYS', 'VEN', 'PAK', 'IRN', 'AFG',
       'MAR', 'DEU', 'ZAF', 'HRV', 'COL', 'ISR', 'EGY', 'PRI', 'POL',
```

```
       'SRB', 'IRL', 'PAN', 'PHL', 'CRI', 'HND', 'SWE', 'ITA', 'NLD',
       'DOM', 'ARG', 'CYP', 'BGD', 'JAM', 'SYR', 'KWT', 'ROU', 'UKR',
       'BGR', 'SVK', 'TUN', 'DZA', 'IRQ', 'FIN', 'CS-KM', 'JPN', 'LBN',
       'MDG', 'LKA', 'BIH', 'LVA', 'SUR', 'GTM', 'UZB', 'GHA', 'BRA',
       'NPL', 'BOL', 'PRT', 'QAT', 'JOR', 'SAU', 'TUR', 'MMR', 'DNK',
       'LUX', 'HUN', 'MLT', 'ARE', 'CUB', 'ZWE', 'AZE', 'FJI', 'SLB',
       'EST', 'OMN', 'LSO', 'NGA', 'CHE', 'KEN', 'HTI', 'GUY', 'PER',
       'CZE', 'TTO', 'BLZ', 'ECU', 'SLV', 'KOR', 'NAM', 'ZMB', 'KHM',
       'AGO', 'ETH', 'CMR', 'TLS', 'MKD', 'ALB', 'SEN', 'BWA', 'GAB',
       'URY', 'KAZ', 'MNG', 'LAO', 'LBY'], dtype=object)
```

In [67]:

```
# Ensure latitude and longitude are within valid ranges
ufo_df = ufo_df[(ufo_df['latitude'] >= -90) & (ufo_df['latitude'] <= 90)]
ufo_df = ufo_df[(ufo_df['longitude'] >= -180) & (ufo_df['longitude'] <= 18

# Verify the changes
ufo_df[['latitude', 'longitude']].describe()
```

Out[67]:

	latitude	longitude
count	79413.000000	79413.000000
mean	38.242312	-87.747623
std	10.080993	37.669803
min	-46.413187	-176.658056
25%	34.149287	-112.166111
50%	39.433611	-87.992268
75%	42.779444	-78.886944
max	72.700000	178.441900

Grouping UFO Shape Sizes

In [68]:

```
# Standardize the UFO shape
# Convert 'UFO_shape' to lowercase for consistency
ufo_df['UFO_shape'] = ufo_df['UFO_shape'].str.lower().str.strip()

# Map similar shapes to a single standardized shape
shape_mapping = {
    'circle': 'circular',
    'light': 'light',
    'cylinder': 'cylinder',
    # Add more mappings as needed
}
ufo_df['UFO_shape'] = ufo_df['UFO_shape'].map(shape_mapping).fillna(ufo_d

# Verify changes
ufo_df['UFO_shape'].unique()
```

Out[68]:

```
array(['cylinder', 'light', 'circular', 'sphere', 'disk', 'fireball',
       'unknown', 'oval', 'other', 'cigar', 'rectangle', 'chevron',
       'triangle', 'formation', 'delta', 'changing', 'egg', 'diamond',
       'flash', 'teardrop', 'cone', 'cross', 'pyramid', 'round',
       'crescent', 'flare', 'hexagon', 'dome', 'changed'], dtype=object)
```

Data Analysis

The following analysis explores the global patterns of UFO sightings, including geographic, temporal, and descriptive aspects. We aim to answer three key business questions using visualizations and statistical analysis.

UFO Sightings by Regions and Time

We will use the Country, Region, Year, Month, Hour, and Season columns to determine the hotspots and peak times for UFO activity.

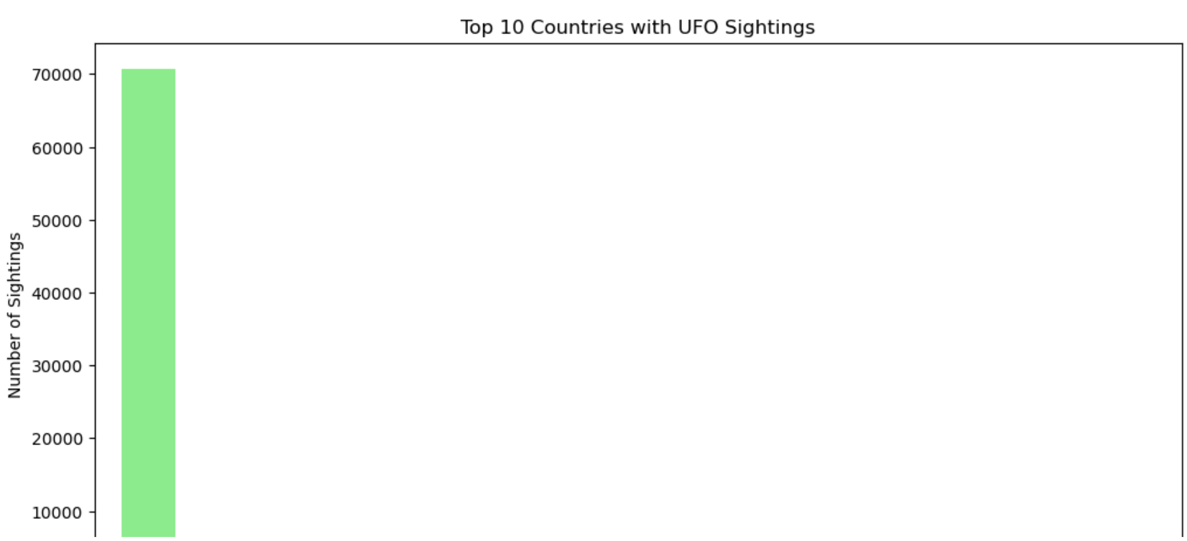
In [69]:

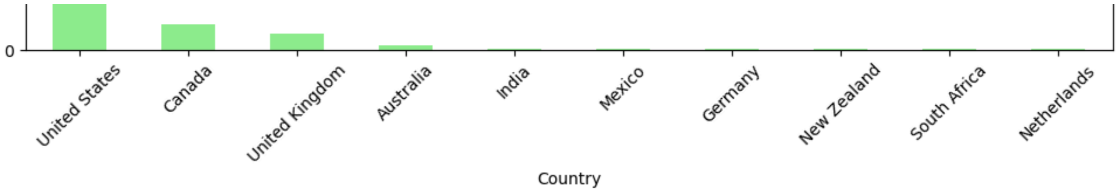
```
#Frequency of sightings by Country
sightings_by_country = ufo_df['Country'].value_counts().head(10)  # Displa
print(sightings_by_country)
```

```
Country
United States     70727
Canada             3546
United Kingdom     2303
Australia           559
India               215
Mexico              209
Germany             116
New Zealand          94
South Africa         94
Netherlands          92
Name: count, dtype: int64
```

In [70]:

```
# Plot the frequency of sightings by country
plt.figure(figsize=(12, 6))
sightings_by_country.plot(kind='bar', color='lightgreen')
plt.title('Top 10 Countries with UFO Sightings')
plt.xlabel('Country')
plt.ylabel('Number of Sightings')
plt.xticks(rotation=45)
plt.show()
```





The United States overwhelmingly leads in UFO sightings, accounting for the vast majority of reports globally, followed by Canada and the United Kingdom. Other countries have significantly fewer sightings, indicating potential reporting or visibility factors.

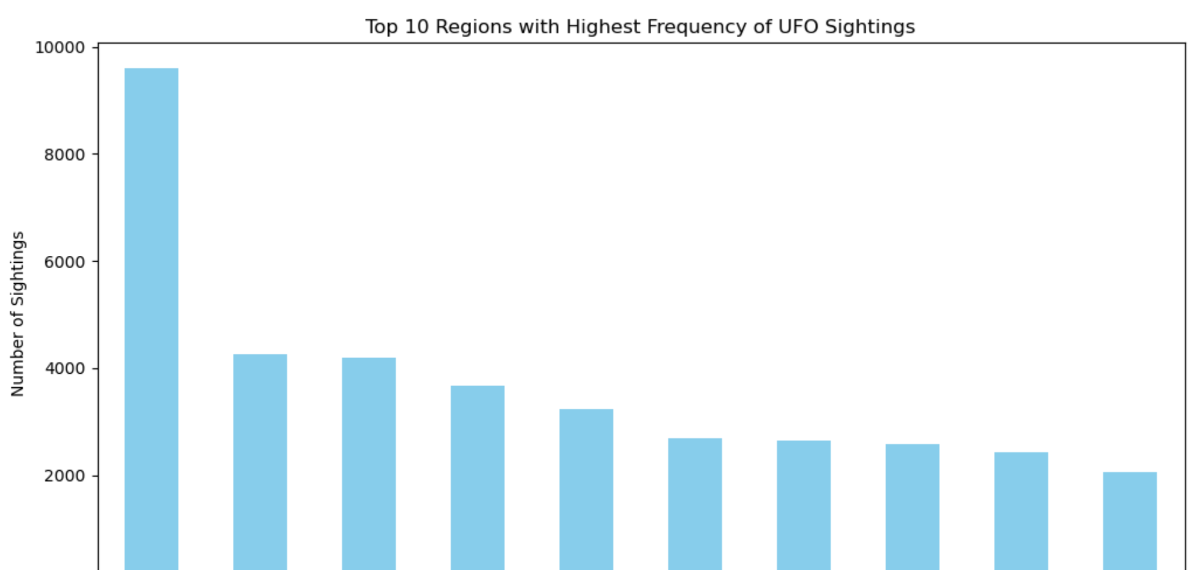
In [71]:

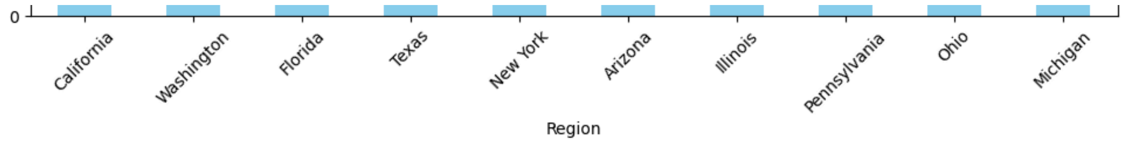
```
# Frequency of sightings by region
region_counts = ufo_df['Region'].value_counts().head(10)  # Top 10 region:
print(region_counts)
```

```
Region
California      9597
Washington      4263
Florida         4196
Texas           3677
New York        3232
Arizona         2680
Illinois        2647
Pennsylvania    2573
Ohio            2423
Michigan        2052
Name: count, dtype: int64
```

In [72]:

```
# Plotting the top regions
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
region_counts.plot(kind='bar', color='skyblue')
plt.title('Top 10 Regions with Highest Frequency of UFO Sightings')
plt.xlabel('Region')
plt.ylabel('Number of Sightings')
plt.xticks(rotation=45)
plt.show()
```





California has the highest number of UFO sightings among U.S. states, followed by Washington and Florida. These regional hotspots suggest that coastal and densely populated states may be more prone to UFO sightings.

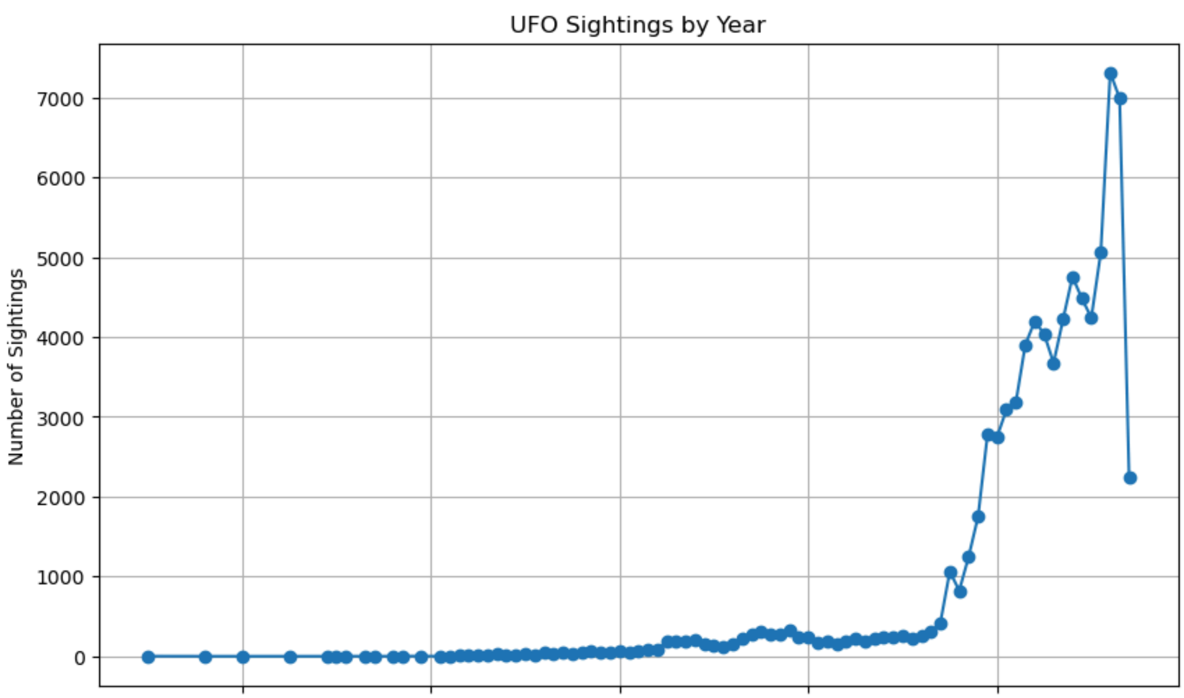
In [73]:

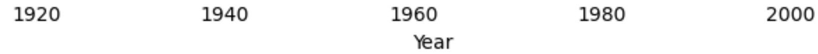
```
#Frequency of sightings by Year
sightings_by_year = ufo_df['Year'].value_counts().sort_index()
print(sightings_by_year)
```

```
Year
1910       2
1916       1
1920       1
1925       1
1929       1
        ...
2010    4241
2011    5058
2012    7312
2013    6988
2014    2244
Name: count, Length: 86, dtype: int64
```

In [74]:

```
# Plot the frequency of sightings over the years
plt.figure(figsize=(10, 6))
plt.plot(sightings_by_year.index, sightings_by_year.values, marker='o')
plt.title('UFO Sightings by Year')
plt.xlabel('Year')
plt.ylabel('Number of Sightings')
plt.grid(True)
plt.show()
```



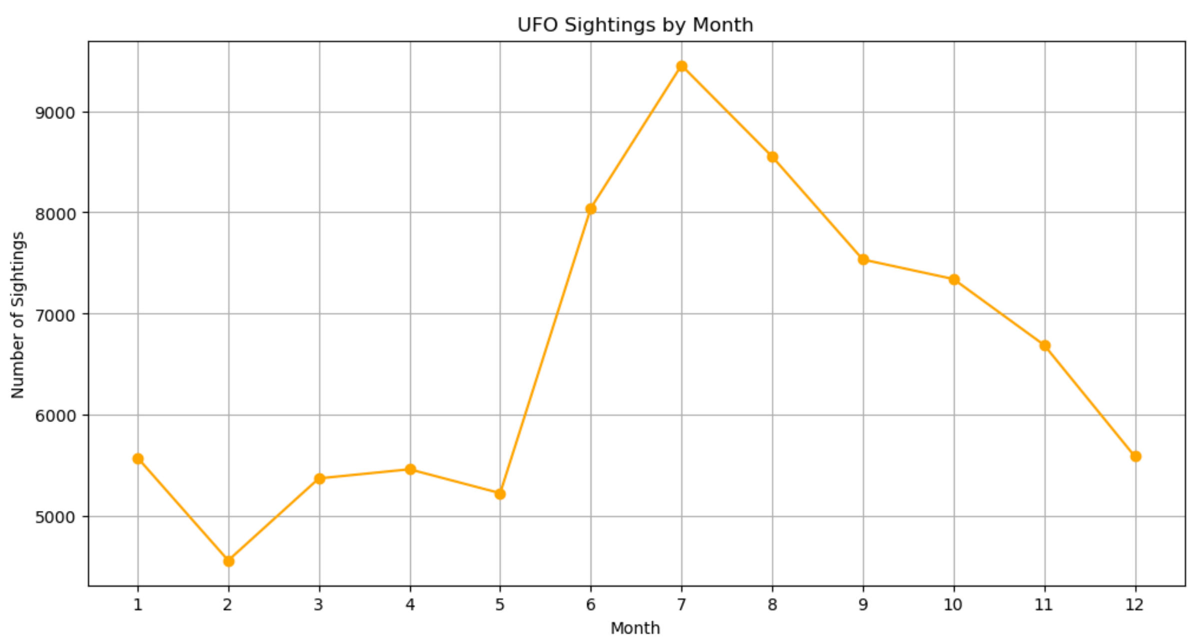


There is a notable rise in UFO sightings starting from 2010, peaking in 2012, which could be linked to increased public awareness or changes in reporting mechanisms. The decline after 2014 may suggest a stabilization or decrease in interest.

In [75]:
```
# Frequency of sightings by time (Month)
monthly_counts = ufo_df['Month'].value_counts().sort_index()
print(monthly_counts)
```

```
Month
1     5578
2     4564
3     5374
4     5464
5     5228
6     8042
7     9452
8     8551
9     7535
10    7342
11    6691
12    5592
Name: count, dtype: int64
```

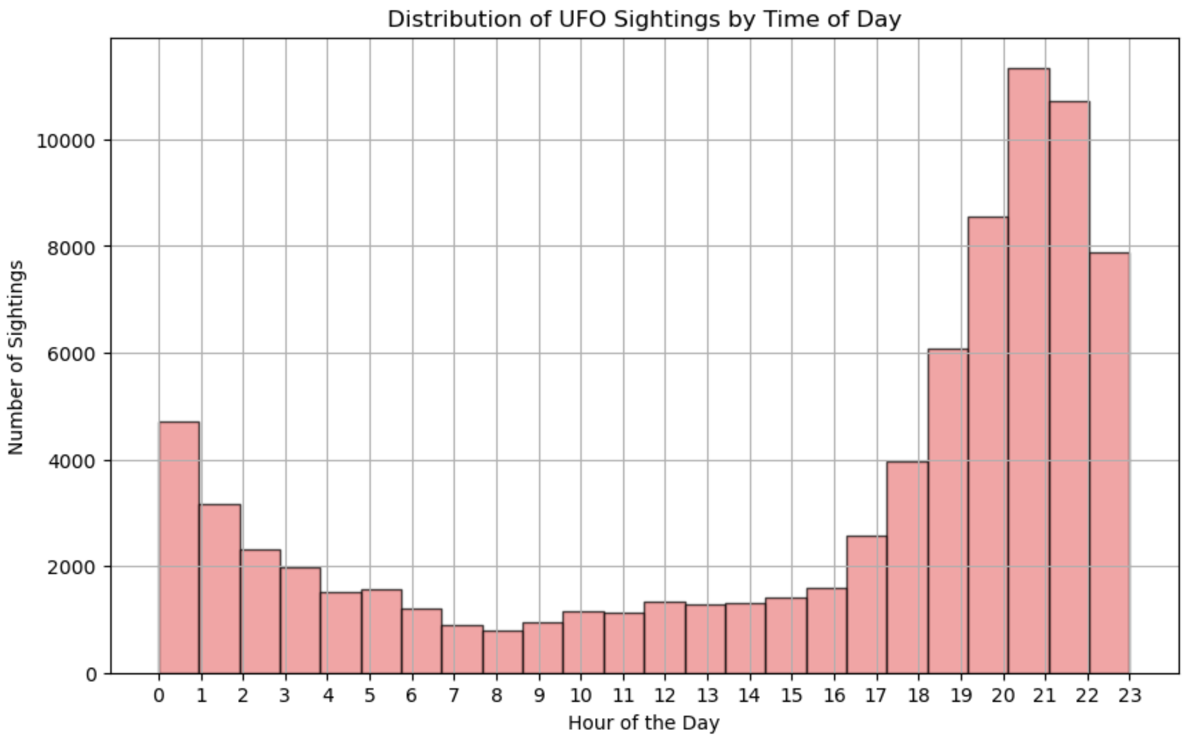
In [76]:
```
# Plotting the monthly sightings
plt.figure(figsize=(12, 6))
monthly_counts.plot(kind='line', marker='o', color='orange')
plt.title('UFO Sightings by Month')
plt.xlabel('Month')
plt.ylabel('Number of Sightings')
plt.xticks(range(1, 13))
plt.grid()
plt.show()
```



July and August show the highest frequency of UFO sightings, with summer months generally leading, while sightings taper off during the winter. This suggests a potential seasonal pattern, possibly influenced by clearer skies or more outdoor activity.

In [77]:

```
#Distribution of sightings by Hour (Time of Day)
plt.figure(figsize=(10, 6))
plt.hist(ufo_df['Hour'], bins=24, color='lightcoral', edgecolor='black', a
plt.title('Distribution of UFO Sightings by Time of Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Number of Sightings')
plt.xticks(range(0, 24))
plt.grid(True)
plt.show()
```



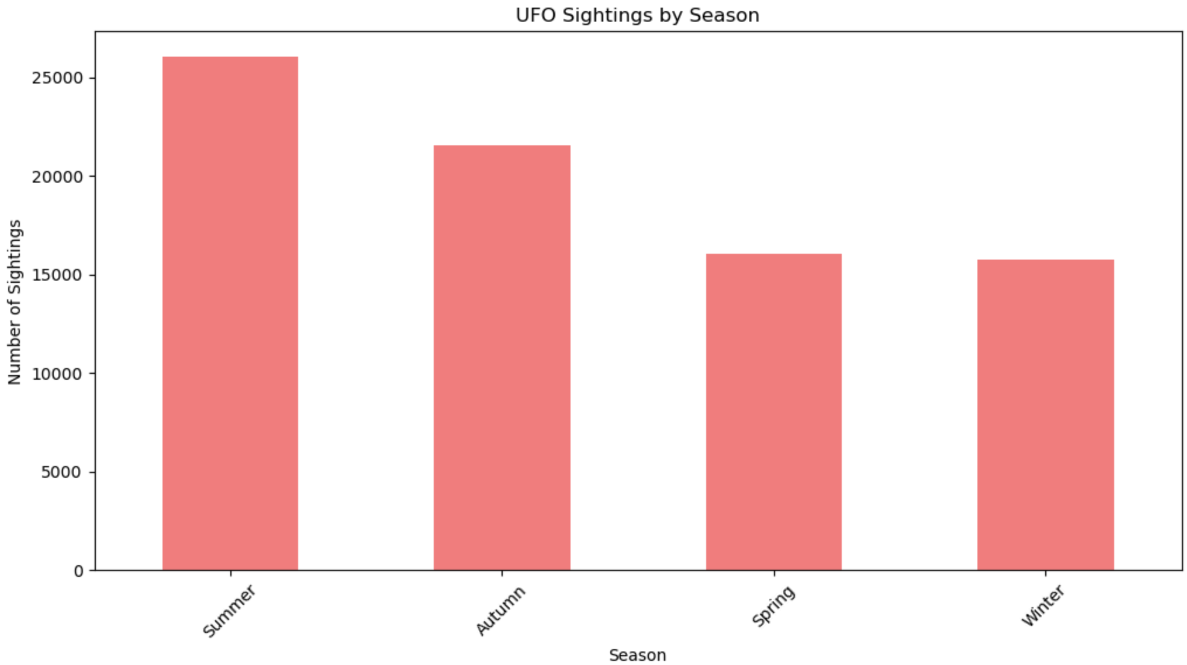
In [78]:

```
# Count sightings by season
season_counts = ufo_df['Season'].value_counts()
print(season_counts)
```

```
Season
Summer    26045
Autumn    21568
Spring    16066
Winter    15734
Name: count, dtype: int64
```

In [79]:

```
# Plotting sightings by season
plt.figure(figsize=(12, 6))
season_counts.plot(kind='bar', color='lightcoral')
plt.title('UFO Sightings by Season')
plt.xlabel('Season')
plt.ylabel('Number of Sightings')
plt.xticks(rotation=45)
plt.show()
```



Summer is the peak season for UFO sightings, followed by autumn. The higher number of sightings during warmer months could be attributed to longer daylight hours and more outdoor visibility opportunities.

Patterns in UFO Shapes and Descriptions

We explore the most frequently reported UFO shapes and lengths of encounters using the UFO_shape and length_of_encounter_seconds columns.

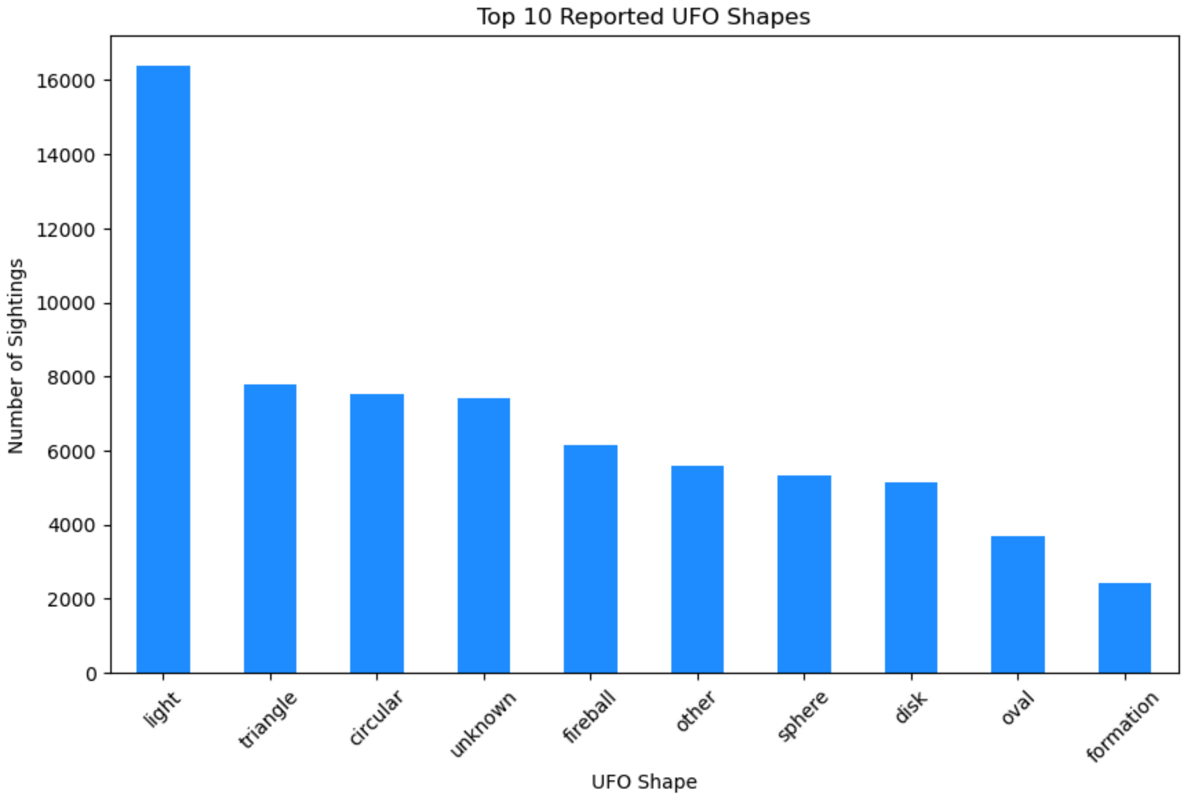
In [80]:

```
#Frequency of different UFO shapes
ufo_shape_counts = ufo_df['UFO_shape'].value_counts().head(10)  # Top 10 r
print(ufo_shape_counts)
```

```
UFO_shape
light        16394
triangle      7798
circular      7515
unknown       7399
fireball      6154
other         5582
sphere        5336
disk          5138
oval          3689
formation     2428
Name: count, dtype: int64
```

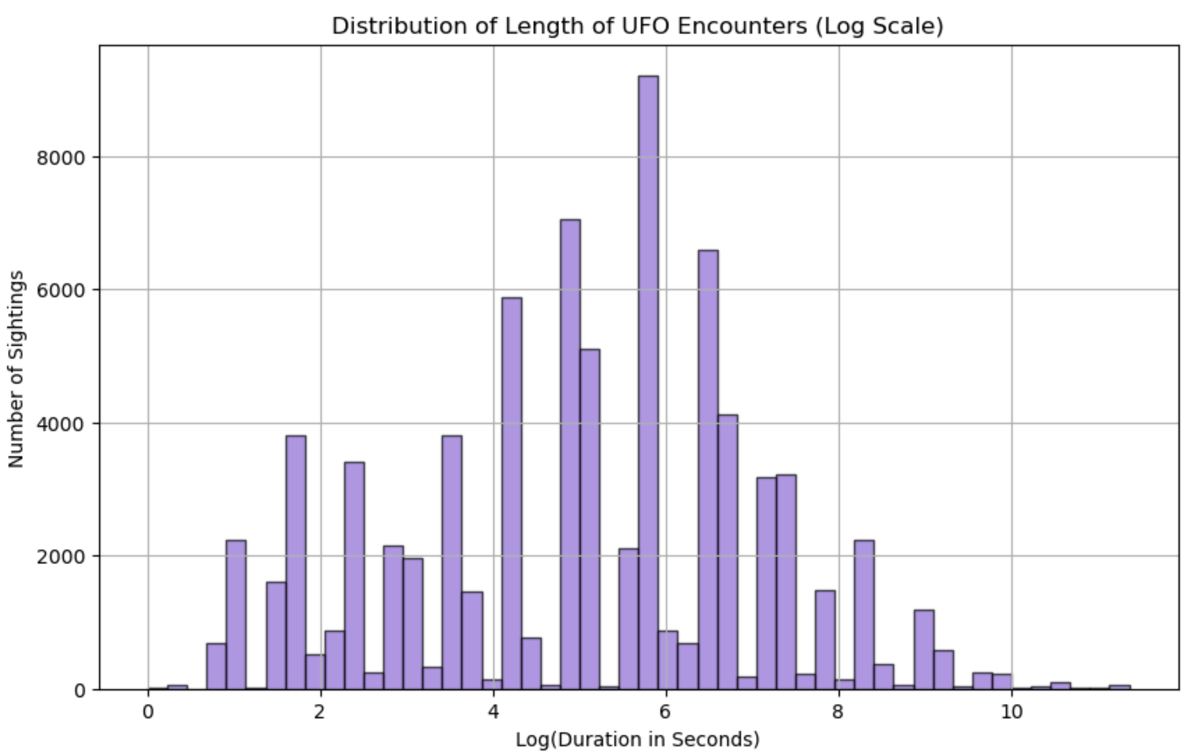
In [81]:

```
# Plot the frequency of UFO shapes
plt.figure(figsize=(10, 6))
ufo_shape_counts.plot(kind='bar', color='dodgerblue')
plt.title('Top 10 Reported UFO Shapes')
plt.xlabel('UFO Shape')
plt.ylabel('Number of Sightings')
plt.xticks(rotation=45)
plt.show()
```



In [82]:

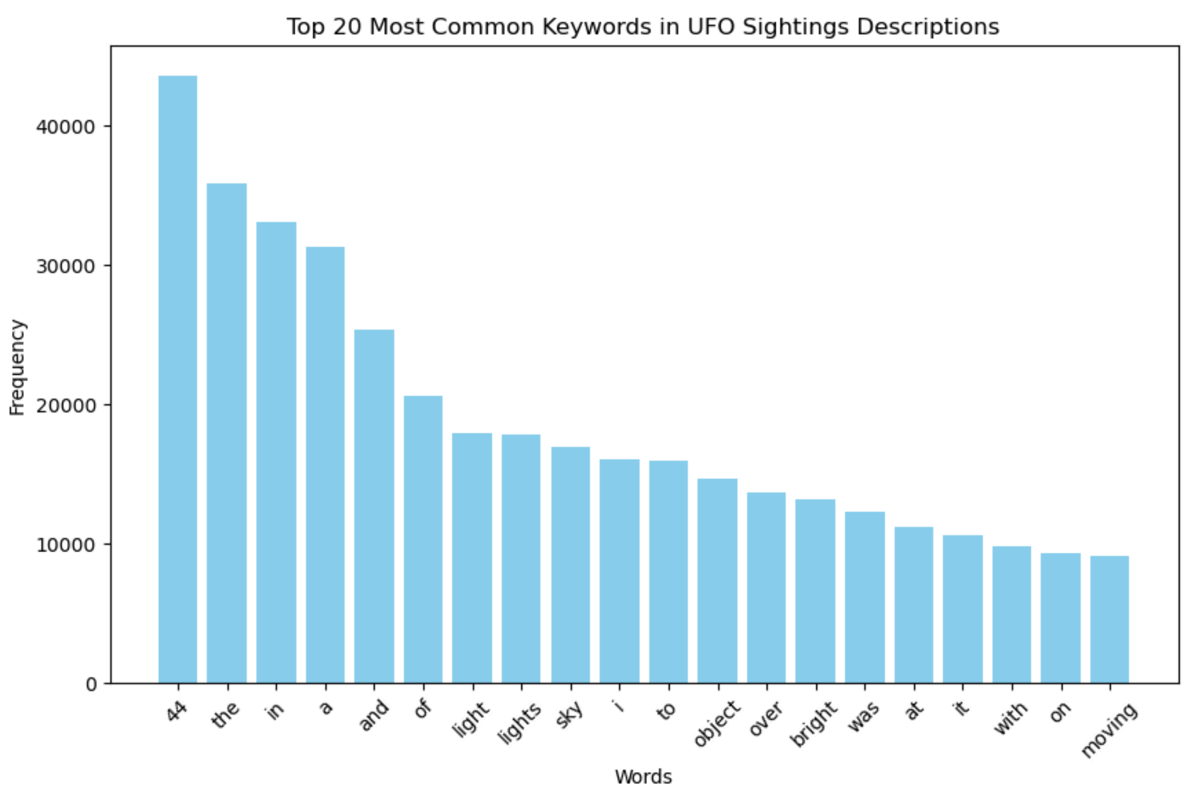
```
#Distribution of length of encounter (log-transformed for better visualiz
plt.figure(figsize=(10, 6))
plt.hist(np.log1p(ufo_df['length_of_encounter_seconds']), bins=50, color=
plt.title('Distribution of Length of UFO Encounters (Log Scale)')
plt.xlabel('Log(Duration in Seconds)')
plt.ylabel('Number of Sightings')
plt.grid(True)
plt.show()
```



Showing that most sightings are relatively short, with a few outliers representing longer encounters.

In [83]:

```
from collections import Counter
import re

# Redefine and extract all words from the 'Description' column
all_descriptions = ' '.join(ufo_df['Description'].astype(str))

# Basic text cleaning: removing special characters and splitting into word
words = re.findall(r'\b\w+\b', all_descriptions.lower())

# Count the frequency of each word
word_counts = Counter(words)

# Get the top 20 most common words
word_counts_most_common = word_counts.most_common(20)
words, counts = zip(*word_counts_most_common)

# Plotting a bar chart for the top 20 most frequent words
plt.figure(figsize=(10, 6))
plt.bar(words, counts, color='skyblue')
plt.title('Top 20 Most Common Keywords in UFO Sightings Descriptions')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



In [84]:

```
pip install nltk
```

```
Requirement already satisfied: nltk in /opt/anaconda3/envs/cohort_env/lib/p
ython3.12/site-packages (3.9.1)
```

```
ython3.12/site-packages (3.9.1)
Requirement already satisfied: click in /opt/anaconda3/envs/cohort_env/lib/
python3.12/site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /opt/anaconda3/envs/cohort_env/li
b/python3.12/site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /opt/anaconda3/envs/cohor
t_env/lib/python3.12/site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in /opt/anaconda3/envs/cohort_env/lib/p
ython3.12/site-packages (from nltk) (4.66.5)
Note: you may need to restart the kernel to use updated packages.
```

In [85]:

```python
import pandas as pd
import nltk
from nltk.corpus import stopwords
from collections import Counter
import matplotlib.pyplot as plt

# Download the stopwords list from nltk
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

# You can also add any additional custom stopwords
custom_stopwords = {"44"}  # Example of custom words to remove
stop_words.update(custom_stopwords)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/saniaspry/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [86]:

```python
# Assuming df['Description'] contains the text data
ufo_df['Description'] = ufo_df['Description'].fillna('').astype(str)

# Tokenize descriptions, remove stopwords, and make all words lowercase
ufo_df['tokens'] = ufo_df['Description'].apply(lambda x: [word.lower() for
```

In [87]:

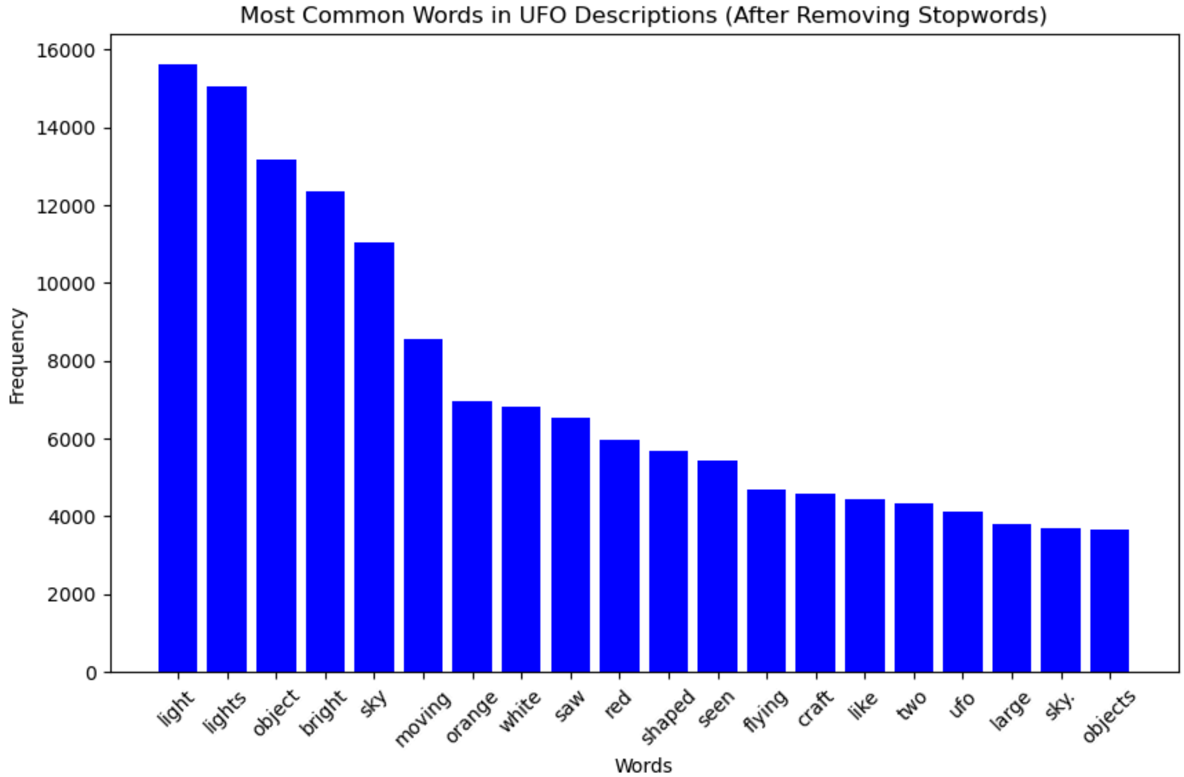
```python
# Flatten the token list to get a list of all words
all_words = [word for tokens in ufo_df['tokens'] for word in tokens]

# Count word frequencies
word_freq = Counter(all_words)

# Get the most common words
most_common_words = word_freq.most_common(20)  # Change the number as need
print(most_common_words)
```

```
[('light', 15612), ('lights', 15045), ('object', 13169), ('bright', 12355),
('sky', 11046), ('moving', 8552), ('orange', 6975), ('white', 6820), ('sa
w', 6523), ('red', 5980), ('shaped', 5679), ('seen', 5426), ('flying', 469
6), ('craft', 4568), ('like', 4443), ('two', 4337), ('ufo', 4137), ('larg
e', 3806), ('sky.', 3692), ('objects', 3651)]
```

In [88]:

```python
words, counts = zip(*most_common_words)
plt.figure(figsize=(10, 6))
plt.bar(words, counts, color='blue')
plt.xticks(rotation=45)
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.title('Most Common Words in UFO Descriptions (After Removing Stopwords
```

```
plt.title('Most Common Words in UFO Descriptions (After Removing Stopwords
plt.show()
```



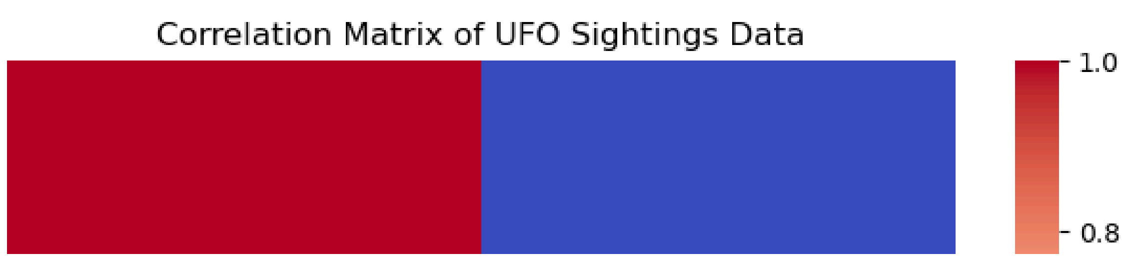
Extracted common keywords in UFO sighting descriptions, offering insights into popular themes and narratives.

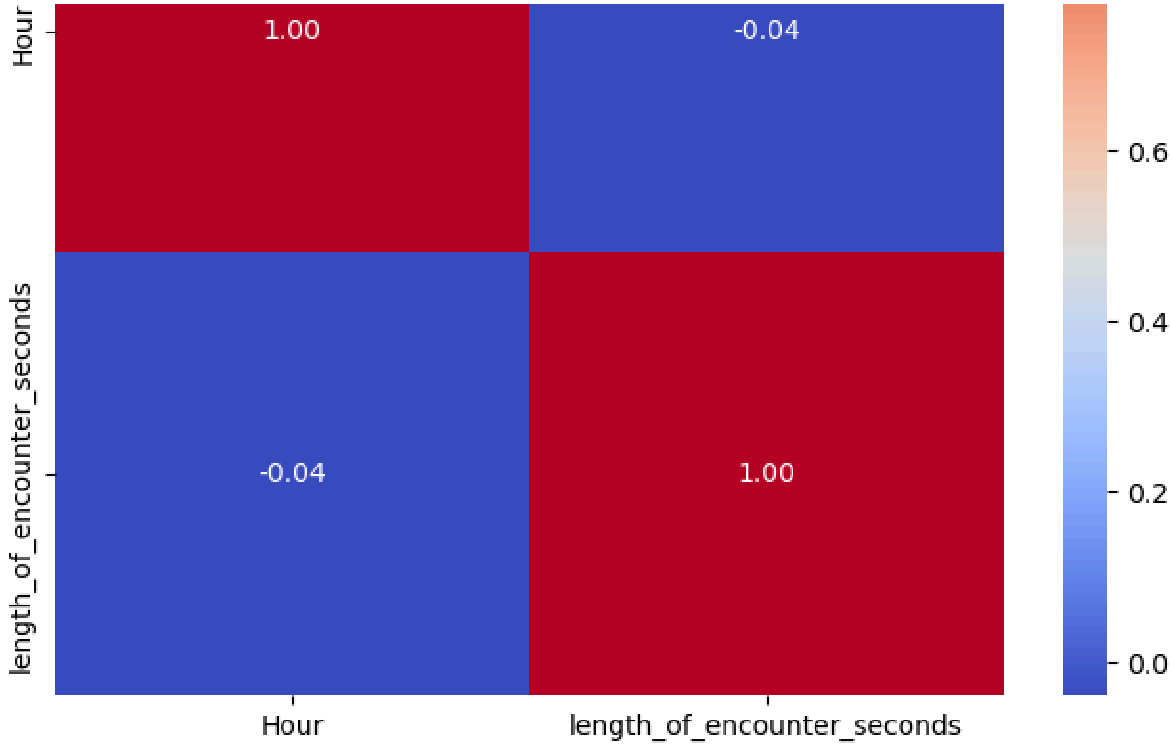
Correlation Between Timing & Sightings

We explore the relationship between timing variables (Season, Month, Hour) and the length of encounters to identify any correlations.

In [89]:

```
# Correlation Analysis using a heatmap
# Selecting numeric columns relevant for correlation analysis
numeric_columns = ['Hour', 'length_of_encounter_seconds']

# Calculate the correlation matrix
correlation_matrix = ufo_df[numeric_columns].corr()

# Plot the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix of UFO Sightings Data')
plt.show()
```

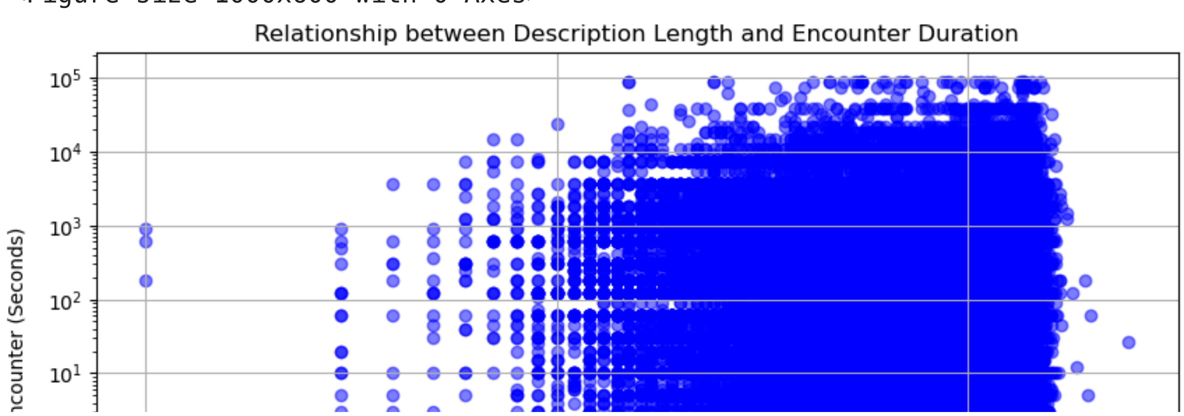


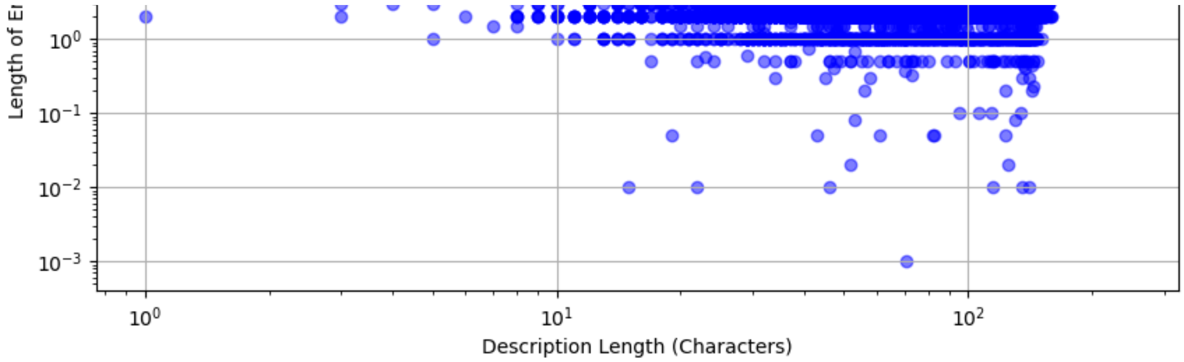


In [92]:

```
# Step 1: Handle NaN values and ensure descriptions are strings
ufo_df['Description'] = ufo_df['Description'].fillna('')  # Replace NaN wi
ufo_df['Description'] = ufo_df['Description'].astype(str)  # Convert to st

# Step 2: Calculate the length of each description (character count)
ufo_df['description_length'] = ufo_df['Description'].apply(len)

# Step 3: Scatter plot of description length vs. length of encounter
plt.figure(figsize=(10, 6))
plt.scatter(ufo_df['description_length'], ufo_df['length_of_encounter_seco
plt.xscale('log')  # Using a log scale if the data is highly skewed
plt.yscale('log')
plt.xlabel('Description Length (Characters)')
plt.ylabel('Length of Encounter (Seconds)')
plt.title('Relationship between Description Length and Encounter Duration
plt.grid(True)
plt.show()

# Step 4: Correlation Analysis
correlation = ufo_df['description_length'].corr(ufo_df['length_of_encount
print(f"Correlation between description length and length of encounter: {
```

<Figure size 1000x600 with 0 Axes>



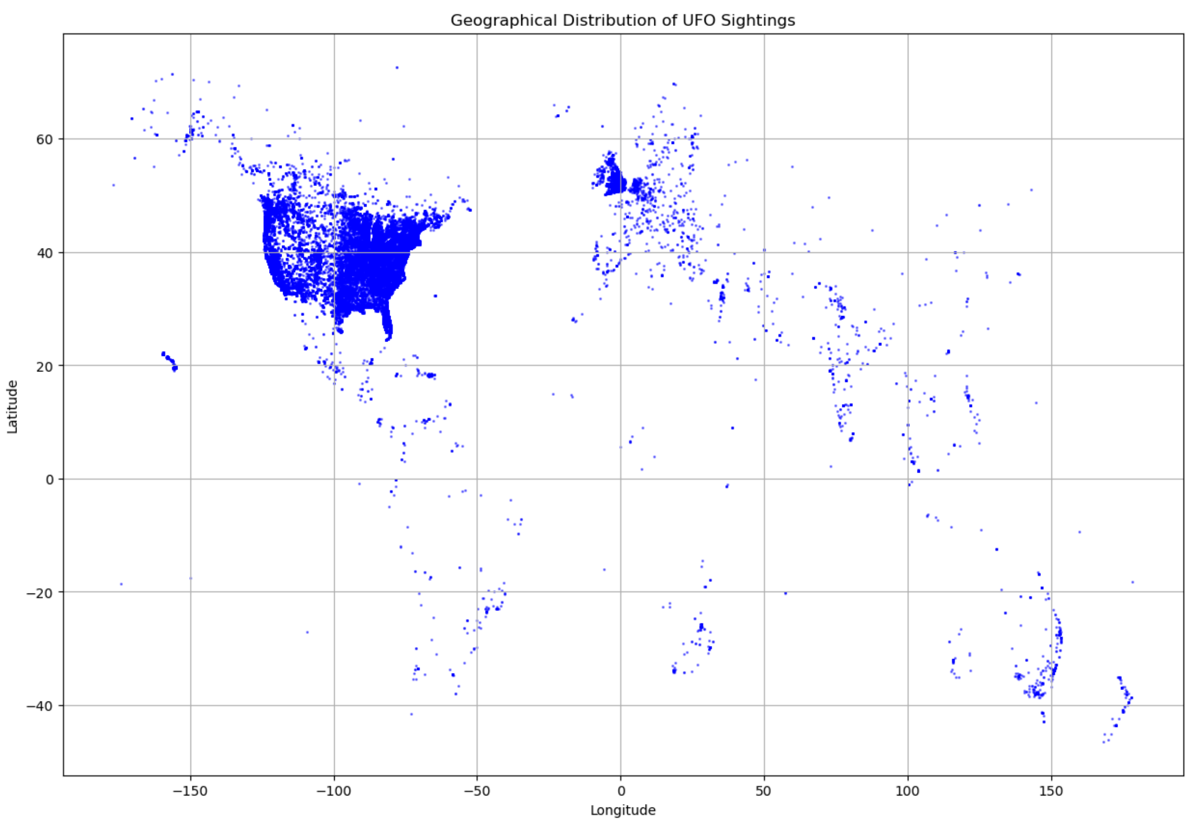



```
Correlation between description length and length of encounter: 0.027539406
911745522
```

It highlights that there is little to no strong correlation between these variables.

Geographic Distribution (Might be better with tableau)

In [90]:
```
# Plotting a scatter plot of UFO sightings using latitude and longitude
plt.figure(figsize=(15, 10))
plt.scatter(ufo_df['longitude'], ufo_df['latitude'], s=1, alpha=0.5, colo
plt.title('Geographical Distribution of UFO Sightings')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.grid(True)
plt.show()
```



The scatter plot displays the geographical distribution of UFO sightings based on latitude and longitude, highlighting regions with higher concentrations of sightings.

Conclusions

- **UFO Sightings by Region and Time**:
  The United States, especially California, leads in UFO sightings by a significant margin. Sightings peak during the summer months, particularly in July and August, and are more frequent in coastal and densely populated areas. The trend over the years shows a rise in sightings after 2010, with a peak in 2012, followed by a decline.

- **Patterns in UFO Shapes and Descriptions**:
  The most commonly reported UFO shape is "light," followed by "triangle" and "circular" forms. The descriptions often include terms like "light," "moving," and "sky." The distribution of the length of encounters suggests most sightings are brief, with a median duration of around 180 seconds.

- **Correlation Between Timing and Sightings**:
  There is no strong correlation between the time of day (hour) and the duration of encounters, with a weak negative correlation (-0.04). This suggests that while there may be certain times with more sightings, the length of these encounters does not significantly vary based on the time.

Additional Insights & Recommendations

- **High Sightings in Certain Regions and at Specific Times of Year**:
  For Tourism Agencies: Create "UFO Tourism Trails" in these high-sighting regions (e.g., California, Washington) with seasonal promotions for autumn and summer. Collaborate with local businesses to offer guided tours during the peak times in the evening (8 p.m. - midnight), focusing on open-sky locations for better visibility.

  For Travel Companies: Design travel packages that include nighttime skywatching experiences in the top sighting regions during the peak months. Add complementary activities like camping, storytelling around sightings, and expert-led UFO talks to enhance the experience.

- **Commonly Reported Shapes and Their Impact on Tourist Experience**:
  For Local Communities: Develop attractions themed around these popular shapes. For example, in regions with a high frequency of "Light" sightings, organize "Light Festivals" featuring sky illuminations, light installations, and educational activities on UFO phenomena.

  For Travel Companies: Create different packages based on shape experiences. "Quick Flash Skywatching" could focus on shorter sightings like "lights" or "circles" with sky-gazing and light shows, while "Deep Sky Mystery Tours" could emphasize longer sightings like "triangles" and "discs," including time for discussions about the sighting descriptions and more in-depth nighttime exploration.

- **Regions with Longer Average Encounter Durations as Key Destinations**:
  For Tourism Agencies & Local Governments: Identify and promote these regions (e.g., states within the U.S. or countries like the UK) as top destinations for in-depth UFO experiences. Create "Extended Sighting Zones" where tourists can experience longer skywatching periods, complete with telescopes, night-vision equipment, and comfortable seating areas.

  For Marketing Teams: Use the unique appeal of these long-duration encounters in advertising campaigns: "Experience the Longest-Lasting UFO Sightings." Highlight local testimonials and stories to enhance the sense of mystery and anticipation for tourists.

- **Emotional Tone of Sightings and Its Connection to Tourist Experience**:
  For Travel Companies: Design tours that cater to different emotional experiences: "Wonder in the Sky" Tours: Focus on long, positive encounters, offering relaxing, awe-inspiring experiences with ambient music, guided sky-gazing, and educational elements.

  "Thrills and Chills Night" Experiences: Leverage sightings described as "scary" or "strange" to create thrill-based tours, including night hikes, scary storytelling, and "mystery" themes to keep tourists on edge.

  For Tour Guides & Educators: Craft narratives around the emotional content of sightings. For sightings with positive sentiments, build a hopeful, awe-inspired story. For sightings with negative sentiment, create suspenseful, thrilling narratives to heighten the tourist experience.

Overall Strategic Enhancements

- Target Key Regions & Timing: Focus on the U.S., especially states like California and Texas, and high-interest regions in the UK, during autumn and summer. Design experiences that are easy to access during peak evening hours.

- Tailor Tourism Packages by Sighting Type: Differentiate packages based on sighting characteristics. Quick encounters can offer "short, bright excitement," while longer sightings can provide in-depth experiences, immersive storytelling, and exploration.

- Highlight Emotional & Unique Experiences: Use sentiment analysis to design tours that connect emotionally with tourists. Play up either the wonder and excitement or the suspense and fear to craft unforgettable UFO-themed adventures.

Next Steps

- **Deeper Analysis of Shape and Duration**:
  Further analysis could explore whether specific shapes, such as "triangle" or "light," correlate with longer or shorter sighting durations, which may provide more insight into the nature of these sightings.

- **Predictive Modeling for Sightings**:
  Building a predictive model that uses variables like time, season, region, and shape to predict future UFO sightings could aid governmental or research bodies in anticipating and preparing for possible events.

- **Anomaly Detection for False Positives**:
  Applying anomaly detection techniques to the dataset could help identify potential "false positives" in the reports, distinguishing between legitimate sightings and potential misinterpretations.

- **Targeted Communication Strategies**:
  Developing communication strategies based on the most common shapes and descriptions could enhance public awareness and provide clearer guidelines for reporting future sightings. This would improve data quality for researchers and policy-makers.