What UFO Sightings Really Mean: A Confluence of Human Behavior and Nature

Statistics 140XP – Fall 2025

Allen Chen Danielle DeFrancisci Manuela Lozano
Debanshi Misra Celine Nugroho Christine Yuan

Table of contents

1 Abstract

UFO sightings are often treated as evidence of unexplained phenomena, but reported observations are shaped just as much by when and where people see things as by what they believe is possible to see. In this project, we analyze over 140,000 public sighting reports from the National UFO Reporting Center (NUFORC), a non-governmental and non-profit organization (Renner 2020). Using both numerical and textual information, we apply statistical methods to model associations between UFO sighting frequency and the political party alignment of each U.S. state. Report counts by state party alignment were modeled using Poisson regression, report rates by state party alignment were modeled using the Wilcoxon Mann-Whitney test, and exploratory visualizations were used to illustrate the temporal and geographic variability of sightings. Our findings show that UFO report frequency differs across state-level political groupings: typically Democratic states tend to be higher-reporting, while typically Republican states tend to be lower-reporting, with swing states falling in the middle ($p < 0.05$). The report rate per state capita did not produce a clear association ($p > 0.05$). Although several environmental and cultural factors may contribute to these differences, at the state level, there appears to be the potential for an association. Further studies and evaluations are required to confirm results. These results highlight the value of understanding how belief, environment, and institutional trust may shape public reporting behavior around UFOs.

2 Introduction

UFO sightings have been a longstanding public fascination intersecting curiosity, imagination, and skepticism. Interest has seemingly grown with the rise of the internet and social media, but most of these sightings have explainable causes. Because individuals often misunderstand what they see, exploring this data can help to understand why certain times or places might have more reported sightings than others. It can also help uncover the difference between

the explainable events and any truly unusual sightings that require more investigation. By analyzing which environmental factors, human beliefs, or behaviors lead to these sightings, we can better understand why UFO sightings happen and why the number of reported sightings might be continuing to grow. This study utilizes public reporting over decades of time, enabling us to observe patterns and potential associations between reporting and social factors such as political parties.

Research on UFO reporting has suggested that belief frameworks, institutional mistrust, and environmental context influence how the public might classify any UFO-like events. Social and political research also shows that belief systems, conspiracy-proneness, and local sociopolitical context shape attention to the interpretation of UFOs: political affiliations and trust in institutions can influence whether ambiguous lights are reported as "UFOs" and how those reports are discussed in media and social networks (French et al. 2008). Prior studies also demonstrate that many UFO reports stem from misinterpretation of environmental factors. Examining why these misinterpretations are made and any commonalities in the individuals reporting these viewings would help uncover why there is still significant attention to this topic. Although our data does not contain the specific demographics of each individual reporter, we do have the geographic information down to the city and state level. The typical political behavior is well known at the state level, so we look to build on past studies by looking specifically at whether political party classification correlates with the number of UFO reportings at the state level.

Existing research shows that many of the UFO sightings have a legitimate environmental or geological explanation, with certain atmospheric or geological phenomena occurring at various times of the day (Medina, Brewer, and Kirkpatrick 2023). For instance, the Marfa Lights are a well-known optical phenomenon where flickering lights are observed and commonly attributed to UFOs, but were disproven by scientists who discovered that it was caused by the interference between car headlights and rising hot air in the desert (Stephan et al. 2009). Thus, the frequency of UFO sightings can also be due to proximity to artificial light sources, such as from planes in airports, cars, and rockets from spaceports.

3 Data Description

3.1 Source

The dataset consists of publicly reported UFO sightings collected by the National UFO Reporting Center (NUFORC). Each sighting includes the time of the event, the city and state, the reported shape of the object, the estimated duration, and a written narrative description. We supplement these records with an external dataset classifying each U.S. state as Democratic, Republican, or Swing(Wikimedia Foundation 2025).

3.2 Structure

Each row represents a single sighting. Variables fall into several categories:

- **Temporal:** raw datetime, and extracted year, month, weekday, and hour
- **Geographic:** city, state
- **Phenomenological:** shape, duration
- **Narrative:** free-text description
- **Political:** state political affiliation
- **Aggregated:** sightings per state

3.3 Key Variables

- `datetime` – original timestamp of the report
- `year`, `month`, `weekday`, `hour` – features extracted for temporal patterns
- `shape` – reported form of the UFO (e.g., light, circle, triangle)
- `duration_seconds` – cleaned and standardized duration
- `description` – written report for NLP analysis
- `party` – political affiliation of the state
- `sightings_per_state` – aggregated state-level totals

In addition to the NUFORC sighting data, we incorporated a separate dataset containing 2020 U.S. Census state population totals. This allowed us to compute per-capita sighting rates and to adjust our statistical models for differences in population size across states. Population was not used as a primary variable, but it served as an important supplemental measure to help interpret reporting frequency more accurately.

3.4 Preprocessing

To prepare the dataset for analysis:

- Missing and unusable entries were removed.
- Datetime strings were cleaned and standardized, and additional temporal features were extracted.
- Duration fields, which were often inconsistently formatted, were cleaned and converted to numeric seconds.
- Narrative descriptions were processed for NLP analysis (lowercasing, tokenization, stopword removal).
- Sighting counts were aggregated at the state level and merged with political affiliation data.

These steps produced a clean, analysis-ready dataset for both statistical modeling and text exploration.

4 Research Question and Hypotheses

This project examines whether sociopolitical context is associated with patterns in reported UFO sightings across the United States.

Primary Research Question:
Does the political affiliation of U.S. states influence the frequency or characteristics of reported UFO sightings?

4.0.1 Hypotheses

- **H1 (Frequency Hypothesis):** UFO sighting counts differ between Democratic, Republican, and Swing states.
- **H2 (Characteristics Hypothesis):** Reported shapes, duration, or narrative sentiment differ across political groups.
- **H0 (Null Hypothesis):** There is no association between political affiliation and UFO sighting frequency or characteristics.

5 Methods

5.1 Study Design

This study uses an observational, retrospective design based on publicly reported UFO sightings. Each state was assigned a political affiliation, and sightings were aggregated by state. Because this is non-experimental, we focus on associations rather than causal claims.

5.2 Statistical Methods

5.2.1 Frequency Comparisons

To evaluate H1, we used several statistical approaches:

- Mann–Whitney U tests (nonparametric)
- Chi-square tests of independence
- A permutation test as a distribution-free robustness check

5.2.2 Count Modeling

Because sightings are count data:

- A **Poisson regression** was used to model sighting counts as a function of political affiliation.
- A **logistic regression** served as a complementary test, predicting state party from sighting count.

5.2.3 Population Normalization

Because states differ dramatically in population size, raw UFO sighting counts partly reflect the number of potential observers. Larger states like California and Texas will naturally report more sightings simply because more people live there. To address this, we incorporated 2020 U.S. Census population totals (U.S. Census Bureau 2021). We created a per-capita sighting measure for each state:

$$\text{sightings per capita} = \frac{\text{sightings per state}}{\text{state population}}$$

This allows us to examine reporting behavior relative to population size rather than relying solely on raw totals. We also fit an additional Poisson regression using population as an offset term:

$$\log(E[\text{sightings}]) = \beta_0 + \beta_1(\text{party}) + \log(\text{population})$$

This model estimates expected sighting rates while adjusting for differences in population. These population-normalized analyses serve as a sensitivity check to make sure the association between political affiliation and reporting frequency is not entirely driven by population size alone.

5.2.4 Characteristics (H2)

To evaluate whether sighting characteristics vary across parties, we applied:

- Chi-square tests for shape distribution
- Mann-Whitney U tests for duration
- Sentiment and text-based comparisons using TF-IDF and clustering

5.2.5 Natural Language Processing

Narrative descriptions were analyzed using:

- TF-IDF vectorization
- K-means clustering
- Sentiment analysis
- Word clouds to highlight frequently used terms

NLP results provide qualitative context to the quantitative analyses.

As noted in the poster results, comparisons of sentiment polarity and descriptive vocabulary between political groups did not show consistent or meaningful differences. This suggests that political alignment affects how often sightings are reported, but not how witnesses describe those sightings.

6 Results

6.1 Reporting Frequency by Political Party

To examine whether political affiliation is associated with sighting volume, states were grouped into high- and low-reporting categories based on the median count. A Chi-square test comparing party vs. sighting level produced:

- $\chi^2 = \mathbf{5.12}$, $\mathbf{df = 1}$, $\mathbf{p = 0.0237}$

This indicates a statistically significant association between political party and reporting level. Democratic states more often fell into the high-reporting category, and Republican states more often fell into the low-reporting group.

6.1.0.1 Contingency Table (party × sight level)

party	High	Low
Dem	17	8
Rep	8	17

A nonparametric Mann–Whitney U test comparing raw counts produced:

- $\mathbf{U = 417.5}$, $\mathbf{p = 0.0426}$

confirming that Democratic states tend to have higher report counts even without distributional assumptions.

The Poisson regression produced a coefficient of approximately 0.5 for Democratic states, indicating a statistically significant increase in expected sighting counts. Interpreted multiplicatively, this corresponds to roughly a 65% higher expected sighting count for Democratic states compared to Republican states, consistent with both the raw and per-capita comparisons.

6.1.1 Per-Capita Reporting Rates

A nonparametric Mann–Whitney U test comparing per capita report rate produced:

- **U = 243, p = 0.1823**

which counters that Democratic states do not have higher report counts even without distributional assumptions.

Because population size strongly influences the total number of sightings a state can generate, we also compared per-capita sighting rates using 2020 Census population data. When we applied a Mann–Whitney U test to compare per-capita reporting rates across political groups, the difference between Democratic and Republican states was not statistically significant. This contrasts with the raw count results and suggests that much of the apparent partisan difference in total sightings is explained by population size rather than by heightened reporting intensity per resident.

Taken together, these results indicate that political affiliation is associated with aggregate reporting volume, but not necessarily with the per-capita rate at which sightings are reported. In other words, Democratic states generate more total reports, but once population is taken into account, the per-resident reporting levels do not differ significantly between political groups.

6.2 Shape Distribution by Political Party

Reported shapes showed a similar pattern across all groups, with "light" being the most common shape nationally. Chi-square tests found no meaningful partisan differences.

6.3 Duration and Bias

Duration was strongly right-skewed, with estimation clusters (e.g., 10, 20, 30 minutes). Mann–Whitney tests showed no systematic differences in duration by party.

6.4 Narrative Text Characteristics

NLP exploration showed:

- Similar vocabulary across parties
- Heavy emphasis on lights, movement, and directional changes
- No large differences in sentiment or theme structure

Thus, political alignment seems to influence *how often* sightings are reported, not *how* they are described.

6.4.1 Narrative and Sentiment Analysis

To examine whether political alignment influences how sightings are described once reported, we analyzed the narrative descriptions associated with each sighting. We first cleaned and tokenized the text, then computed sentiment scores and examined common descriptive terms across political groups.

Across Democratic, Republican, and Swing states, the overall language used in reports was remarkably similar. Most descriptions focused on lights, changes in motion or direction, color, and uncertainty about the object's identity. Sentiment scores also did not differ meaningfully between political groups; the vast majority of reports were neutral in tone and descriptive rather than emotional.

We also explored differences in vocabulary and phrasing using TF-IDF–based comparisons and simple clustering. These analyses did not reveal consistent or substantial distinctions across political groups. Themes such as "bright light," "moving quickly," "hovering," and "disappeared suddenly" appeared uniformly across states regardless of political affiliation.

Taken together, these findings suggest that while political context may be associated with how sightings are reported, it does not appear to influence how sightings are described. Individuals across different political environments tend to frame and narrate their experiences in very similar ways.

6.5 Geographic Distribution

California, Florida, Washington, and Texas accounted for the highest total sightings. These states also have large populations, highlighting that raw counts may partly reflect population size. Future work should weight by population to separate demographic effects from political context.

6.6 Population-Adjusted Results

To understand whether population size was influencing the patterns we observed, we calculated the number of UFO sightings per capita using 2020 Census state population data. States with the highest raw sighting counts also tended to be the states with the largest populations, which is expected, but the per-capita adjustment allowed us to compare reporting behavior on a more even playing field.

After adjusting for population, the general pattern across political groups remained similar. Democratic states still reported more sightings per capita on average than Republican states, although the gap between groups narrowed once population size was taken into account. A Poisson model with a population offset produced consistent results, suggesting that population is a contributing factor but does not explain the association on its own.

These findings indicate that while population size shapes the total number of sightings, political affiliation still appears to be associated with reporting rates even after controlling for the number of potential observers in each state.

7 Discussion

This study examined whether U.S. state political affiliation is associated with UFO sighting frequency. Results clearly support H1: UFO report volume differs across political groups. Democratic states tend to report more sightings than Republican states, a finding supported across multiple statistical tests and Poisson modeling. However, once sightings were normalized by state population, the difference between political groups was no longer statistically significant. This suggests that part of the observed disparity in total sightings reflects differences in population size rather than higher reporting intensity per resident. Even so, the aggregate patterns still point to meaningful differences in overall reporting behavior that may relate to cultural attitudes, civic engagement, or urbanization patterns.

In contrast, H2 was not supported. Shape distributions, sighting durations, and narrative sentiment were similar across political groups, suggesting that political context shapes reporting frequency but not the nature of the reports themselves. Individuals across the political spectrum tend to describe similar kinds of UFO features using similar language, indicating that the qualitative aspects of UFO sightings are relatively stable regardless of political environment.

Several limitations should be noted: the data are self-reported, state-level political labels are coarse, and environmental visibility variables were not included. NLP tools may miss subtle narrative differences, and the lack of population-adjusted significance highlights the importance of controlling for demographic factors in future work. Incorporating finer-grained political measures—such as county-level patterns or urban–rural splits—may also provide additional insight into how sociopolitical context interacts with public reporting.

Even with these limitations, the results add to our understanding of how sociopolitical context intersects with public reporting of unusual aerial events. They show that people across the political spectrum describe similar experiences, but differ in how often they choose to report them, and that population size plays an important role in shaping these aggregate patterns.

8 Conclusion

This study provides evidence that UFO sighting frequency differs across political contexts in the United States, with Democratic states exhibiting higher reporting levels than Republican states in raw counts and Poisson modeling. However, after adjusting for state population, per-capita reporting rates did not differ significantly, suggesting that some of the observed disparity reflects underlying demographic patterns rather than higher reporting intensity per resident. At the same time, the qualitative characteristics of sightings showed no strong systematic differences by political affiliation. Thus, political context appears to influence the likelihood of reporting UFOs more than the nature of the phenomena described.

These findings underscore the importance of considering sociopolitical and demographic factors when interpreting public UFO sighting databases. While political affiliation should not be construed as a causal determinant of sightings, it may serve as a proxy for broader cultural or behavioral influences on reporting practices. Future research should incorporate finer-grained political measures, environmental visibility variables, and additional demographic controls to more precisely characterize drivers of UFO reporting behavior. More advanced NLP techniques may also yield deeper insights into subtle linguistic patterns across political or regional groups.

Overall, this study demonstrates the value of integrating traditional statistical methods with natural language processing to investigate social and cultural patterns in public reporting data. By combining frequency-based analyses with narrative characteristics, we gain a richer understanding of how individuals across the United States perceive and describe unusual aerial events, and how these patterns intersect with political, environmental, and demographic contexts.

9 Appendix

```
# install.packages("reticulate")  # if not already installed
library(reticulate)

# Create a dedicated environment for this project
virtualenv_create("quarto-eda")
```

```
virtualenv: quarto-eda
```

```r
# Install the Python packages you use in the EDA
virtualenv_install(
  "quarto-eda",
  packages = c("pandas", "numpy", "matplotlib", "seaborn", "wordcloud", "textblob", "spicy",
)
```

```
Using virtual environment "quarto-eda" ...
```

```r
# Tell reticulate to use this env in the current session
use_virtualenv("quarto-eda", required = TRUE)

py_config()  # just to check it sees Python correctly
```

```
python:         C:/Users/chen4/OneDrive/Documents/.virtualenvs/quarto-eda/Scripts/python.exe
libpython:      C:/Data/supermap-iobjectspy-env-gpu-11.2.0-win64/conda/python38.dll
pythonhome:     C:/Users/chen4/OneDrive/Documents/.virtualenvs/quarto-eda
version:        3.8.16 (default, Jun 12 2023, 21:00:42) [MSC v.1916 64 bit (AMD64)]
Architecture:   64bit
numpy:          C:/Users/chen4/OneDrive/Documents/.virtualenvs/quarto-eda/Lib/site-packages/n
numpy_version:  1.24.4

NOTE: Python version was forced by use_python() function
```

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

plt.style.use("seaborn-v0_8")
sns.set_palette("viridis")

# Load raw data (file should be in the same folder as this qmd)

df = pd.read_csv("nuforc2022.csv")

print("Number of rows:", df.shape[0])
```

```
Number of rows: 141261
```

```
print("Number of columns:", df.shape[1])
```

```
Number of columns: 11
```

```
df.head()
```

```
                                                text  ...   posted
0                                     MADAR Node 100  ...  6/27/19
1  Steady flashing object with three lights hover...  ...  6/27/19
2  Group of several orange lights, seemingly circ...  ...  6/27/19
3  Dropped in flashed a few times and shot off 5 ...  ...  6/27/19
4  Location:  While traveling in a TGV, from Lill...  ...  6/27/19

[5 rows x 11 columns]
```

```
# Data Documentation

# Shape of dataset
print("Number of rows:", df.shape[0])
```

```
Number of rows: 141261
```

```
print("Number of columns:", df.shape[1])
```

```
Number of columns: 11
```

```
# Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 141261 entries, 0 to 141260
Data columns (total 11 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   text         141227 non-null  object
 1   stats        141261 non-null  object
 2   date_time    141160 non-null  object
 3   report_link  141261 non-null  object
```

```
 4   city         140796 non-null  object
 5   state        131681 non-null  object
 6   country      140944 non-null  object
 7   shape        134962 non-null  object
 8   duration     133642 non-null  object
 9   summary      141189 non-null  object
 10  posted       141261 non-null  object
dtypes: object(11)
memory usage: 11.9+ MB
```

```
# Data types
df.dtypes
```

```
text           object
stats          object
date_time      object
report_link    object
city           object
state          object
country        object
shape          object
duration       object
summary        object
posted         object
dtype: object
```

```
# Unique values per field
unique_counts = df.nunique().sort_values(ascending=False)
print(unique_counts)
```

```
report_link    141261
stats          141247
text           139328
summary        138549
date_time      120710
city            28833
duration        14008
posted            617
country           439
state             297
shape              39
dtype: int64
```

```
# Summary statistics
df.describe(include="all")
```

```
                  text  ...   posted
count           141227  ...   141261
unique          139328  ...      617
top     MADAR Node 119  ...  6/25/20
freq                57  ...     1835

[4 rows x 11 columns]
```

```
# Checking Missing Values

plt.figure(figsize=(10,5))
sns.heatmap(df.isnull(), cbar=False)
plt.title("Missing Values Heatmap")
plt.show()
```


```
df.isnull().sum()
```

```
text             34
stats             0
date_time       101
```

```
report_link       0
city            465
state          9580
country         317
shape          6299
duration       7619
summary          72
posted            0
dtype: int64
```

```
missing_table = pd.DataFrame({
    "missing_count": df.isnull().sum(),
    "missing_percent": (df.isnull().sum() / len(df)) * 100
}).sort_values(by="missing_percent", ascending=False)

missing_table
```

```
             missing_count  missing_percent
state                 9580         6.781773
duration              7619         5.393562
shape                 6299         4.459122
city                   465         0.329178
country                317         0.224407
date_time              101         0.071499
summary                 72         0.050969
text                    34         0.024069
stats                    0         0.000000
report_link              0         0.000000
posted                   0         0.000000
```

```
# Dropping Missing Values

df_clean = df.dropna()

df_clean.shape
```

```
(121662, 11)
```

```
df_clean.isnull().sum()
```

```
text           0
stats          0
date_time      0
report_link    0
city           0
state          0
country        0
shape          0
duration       0
summary        0
posted         0
dtype: int64
```

```
df_clean.tail(30)
```

	text	...	posted
141221	circle with blinking lights. \n \nI was callin...	...	12/19/21
141222	A light in the sky over San Francisco Bay in c...	...	12/19/21
141225	Low rumbling noise then a light. I stood up & ...	...	12/19/21
141226	While taking an photograph of Christmas lights...	...	12/19/21
141227	A light that was fading in and out. \n \nMe an...	...	12/19/21
141228	Multiple lights that turn on off always in the...	...	12/19/21
141229	I saw two bright white lights traveling on the...	...	12/19/21
141231	Two large parallel red lights \n \nI was drivi...	...	12/19/21
141232	Craft seen entering atmosphere \n \nCraft ente...	...	12/19/21
141233	3 fireball-type of objects ascending, travelin...	...	12/19/21
141235	I was recording some shooting stars and ended ...	...	12/19/21
141236	A distant glowing object rapidly descending do...	...	12/19/21
141237	Looking at the geminids and witnessed a format...	...	12/19/21
141239	Object was Pretending to be a Star \n \nWent o...	...	12/19/21
141240	Object appeared, grew in size and brightness, ...	...	12/19/21
141241	I saw a sphere of light shoot through the clou...	...	12/19/21
141242	Silent triangular craft about 75 yrds above ro...	...	12/19/21
141243	Accidental capture of disc-shaped UFO in photo...	...	12/19/21
141244	At the stop sign look up so three lights that ...	...	12/19/21
141246	I saw a orb like light changing colors and mov...	...	12/19/21
141247	Witnessed a large silver light in the sky alon...	...	12/19/21
141248	It was loud like rocket in a V shape. \n \nWhe...	...	12/19/21
141249	I saw a green meteor like thing with a tail fa...	...	12/19/21
141250	2 ufos hovering in my backyard, 40-50ft high \...	...	12/19/21
141251	Silent craft flew overhead at extremely high s...	...	12/19/21
141253	"Fireball" of green light, vertically descendi...	...	12/19/21

```
141254  Two lights spinning around each other . See vi...  ...  12/19/21
141258  A very small white light hovering above the cl...  ...   5/15/13
141259  I was young.  You know what?  It was pretty ((...  ...  12/19/19
141260  While driving at night, I watched two blue-gre...  ...   5/14/19

[30 rows x 11 columns]
```

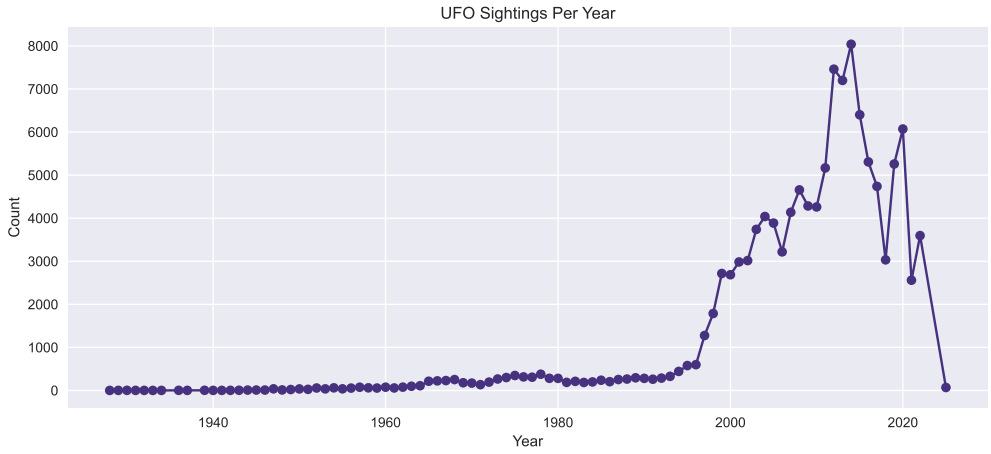
```
# EDA

# Date/Time Cleaning

df_clean["date_time"] = pd.to_datetime(
    df_clean["date_time"],
    errors="coerce",
    yearfirst=False,
    infer_datetime_format=True
)


# After parsing date_time as above
df_clean["year"] = df_clean["date_time"].dt.year

# Look at suspicious future years
print(df_clean["year"].max())
```

```
2074.0
```

```
# Treat anything after 2025 as mis-entered and shift back 100 years
mask_future = df_clean["year"] > 2025
df_clean.loc[mask_future, "date_time"] = df_clean.loc[mask_future, "date_time"] - pd.DateOffs

# Recompute year, month, weekday
df_clean["year"] = df_clean["date_time"].dt.year
df_clean["month"] = df_clean["date_time"].dt.month
df_clean["weekday"] = df_clean["date_time"].dt.day_name()

df_clean["year"].agg(["min", "max"])
```

```
min    1928.0
max    2025.0
Name: year, dtype: float64
```

```
# Sights Over Time

# Sights per Year
plt.figure(figsize=(12,5))
df_clean.year.value_counts().sort_index().plot(kind="line", marker="o")
plt.title("UFO Sightings Per Year")
plt.xlabel("Year")
plt.ylabel("Count")
plt.grid(True)
plt.show()
```



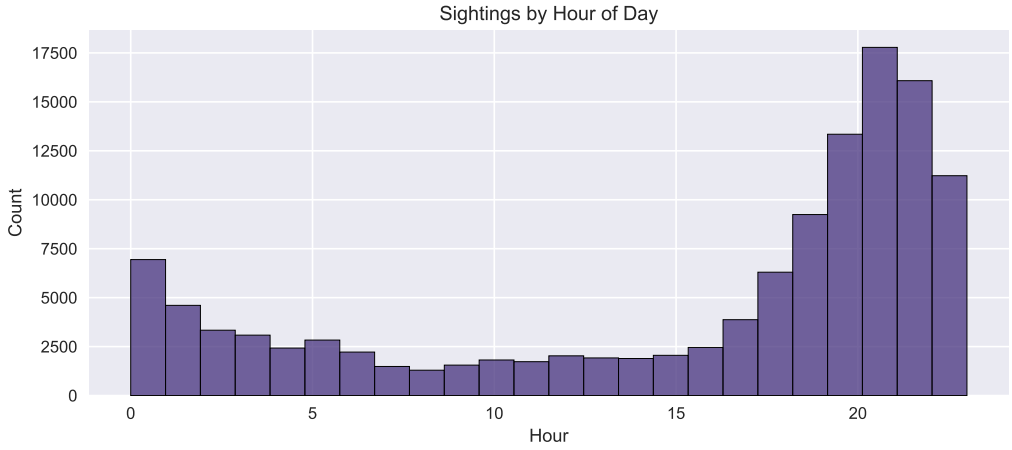

```
# Sights per Month
plt.figure(figsize=(12,5))
df_clean.month.value_counts().sort_index().plot(kind="line", marker="o")
plt.title("Sightings by Month")
plt.xlabel("Month")
plt.ylabel("Count")
plt.show()
```


```
# Sights by Weekday
plt.figure(figsize=(10,4))
sns.countplot(x="weekday", data=df_clean, order=[
    "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"
])
plt.title("Sightings by Weekday")
plt.xticks(rotation=45)
```

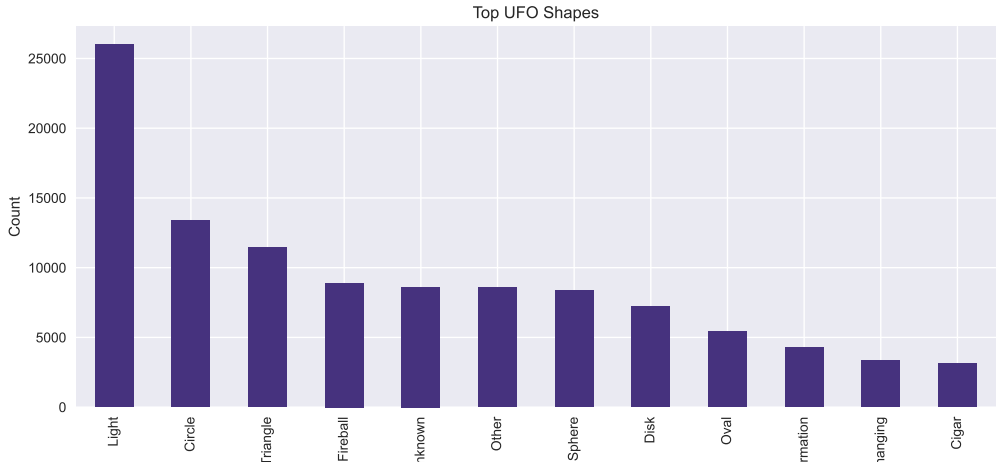
```
([0, 1, 2, 3, 4, 5, 6], [Text(0, 0, 'Monday'), Text(1, 0, 'Tuesday'), Text(2, 0, 'Wednesday'
```

```
plt.show()
```

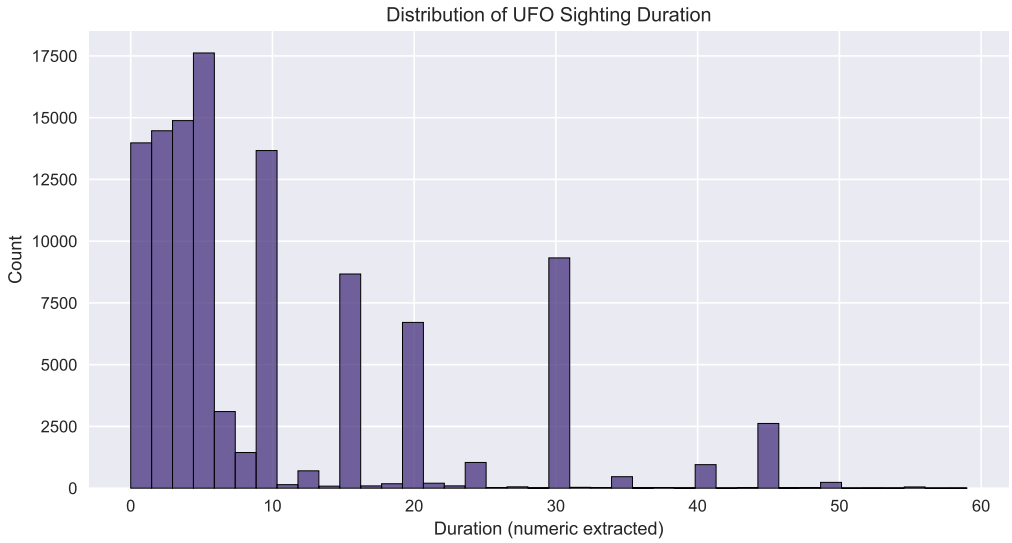
```
# Hourly Pattern
df_clean["hour"] = df_clean["date_time"].dt.hour

plt.figure(figsize=(10,4))
sns.histplot(df_clean["hour"], bins=24)
plt.title("Sightings by Hour of Day")
plt.xlabel("Hour")
plt.show()
```




```
# UFO Shape Distribution
plt.figure(figsize=(12,5))
df_clean["shape"].value_counts().head(12).plot(kind="bar")
plt.title("Top UFO Shapes")
plt.xlabel("Shape")
plt.ylabel("Count")
plt.show()
```




```
# Duration Cleaning
df_clean["duration_numeric"] = (
    df_clean["duration"]
    .astype(str)
    .str.extract(r"(\d+\.?\d*)")
)

df_clean["duration_numeric"] = pd.to_numeric(df_clean["duration_numeric"], errors="coerce")

df_clean = df_clean.dropna(subset=["duration_numeric"])

# Removing extreme outliers
df_no_outliers = df_clean[df_clean["duration_numeric"] < df_clean["duration_numeric"].quantil

# Plot duration distribution
plt.figure(figsize=(10,5))
sns.histplot(df_no_outliers["duration_numeric"], bins=40)
plt.title("Distribution of UFO Sighting Duration")
plt.xlabel("Duration (numeric extracted)")
plt.show()
```



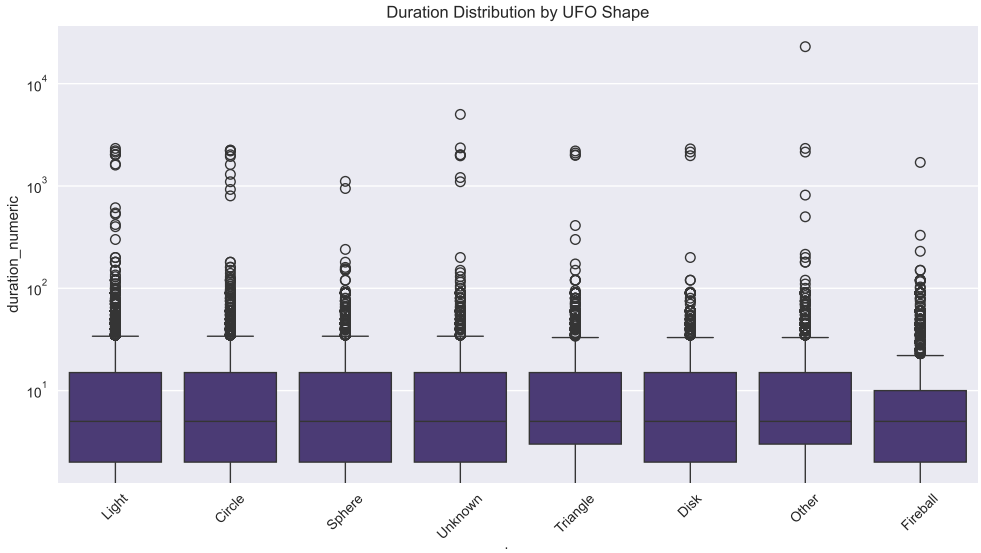

```
# Duration vs. Shape (Variability Analysis)
# 1. Make sure duration_numeric exists on df_clean
df_clean = df_clean.copy()

df_clean["duration_numeric"] = (
    df_clean["duration"]
    .astype(str)
    .str.extract(r"(\d+\.?\d*)")[0]  # take first capture group
)

df_clean["duration_numeric"] = pd.to_numeric(df_clean["duration_numeric"], errors="coerce")

# 2. Drop missing duration
df_clean = df_clean.dropna(subset=["duration_numeric"])

# 3. Pick top shapes and create subset
top_shapes = df_clean["shape"].dropna().value_counts().head(8).index

subset = df_clean[
    df_clean["shape"].isin(top_shapes)
    & df_clean["duration_numeric"].notna()
]

print(subset.columns)  # quick check that duration_numeric is there
```

```
Index(['text', 'stats', 'date_time', 'report_link', 'city', 'state', 'country',
       'shape', 'duration', 'summary', 'posted', 'year', 'month', 'weekday',
       'hour', 'duration_numeric'],
      dtype='object')
```

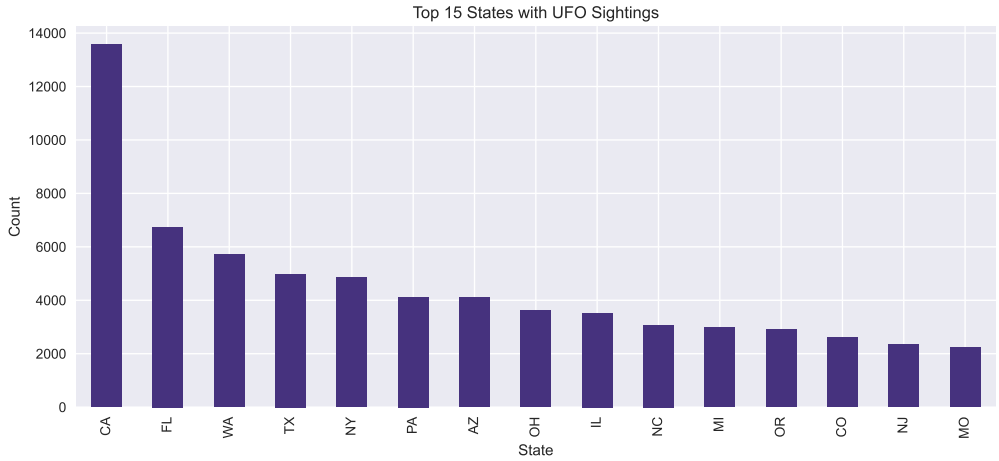
```
# 4. Boxplot: duration vs shape
plt.figure(figsize=(12,6))
sns.boxplot(data=subset, x="shape", y="duration_numeric")
plt.title("Duration Distribution by UFO Shape")
plt.xticks(rotation=45)
```

```
([0, 1, 2, 3, 4, 5, 6, 7], [Text(0, 0, 'Light'), Text(1, 0, 'Circle'), Text(2, 0, 'Sphere'),
```

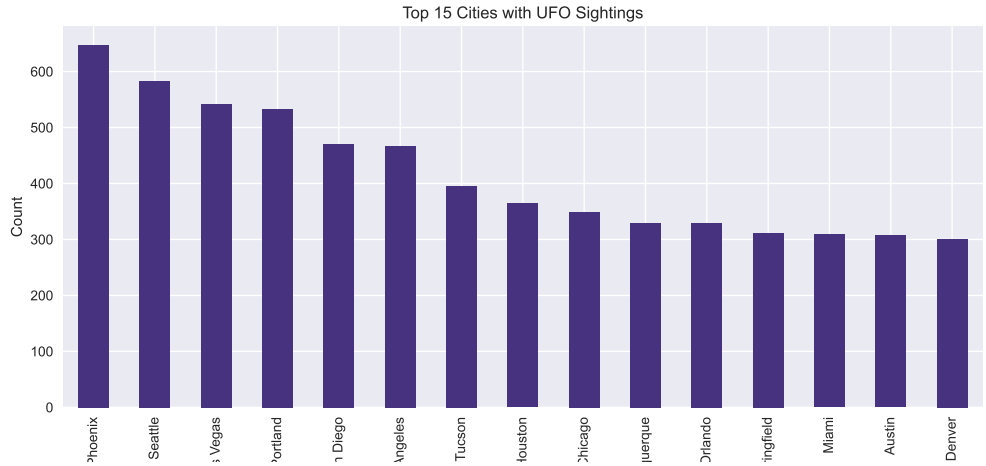
```
plt.yscale("log")  # optional, because it's very skewed
plt.show()
```




```
# Cities / States with Most Sightings
# Top 15 states
plt.figure(figsize=(12,5))
df_clean.state.value_counts().head(15).plot(kind="bar")
plt.title("Top 15 States with UFO Sightings")
plt.xlabel("State")
plt.ylabel("Count")
plt.show()
```




```
# Top 15 cities
plt.figure(figsize=(12,5))
df_clean.city.value_counts().head(15).plot(kind="bar")
plt.title("Top 15 Cities with UFO Sightings")
plt.xlabel("City")
plt.ylabel("Count")
plt.show()
```

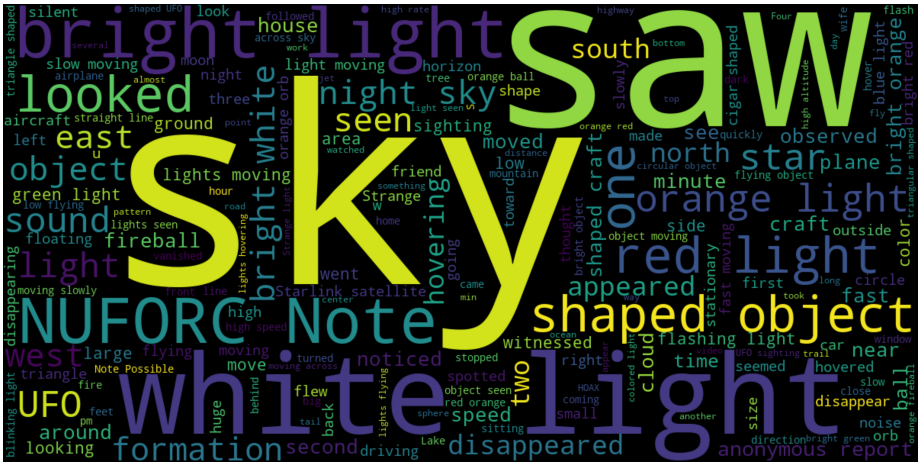



```
# Word Cloud
from wordcloud import WordCloud

text = " ".join(df_clean["summary"].dropna())
```

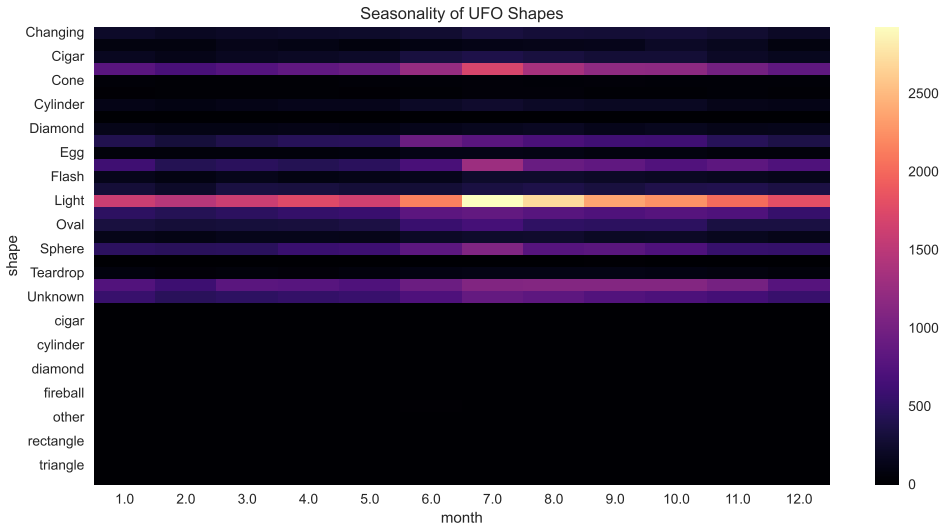
```
wc = WordCloud(width=1200, height=600, background_color="black").generate(text)

plt.figure(figsize=(12,6))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
```

```
(-0.5, 1199.5, 599.5, -0.5)
```

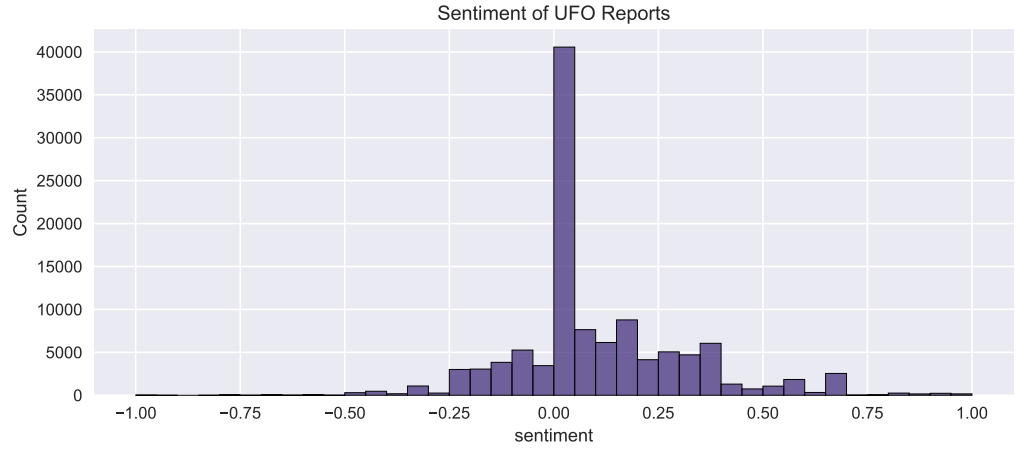
```
plt.show()
```




```
# Shape vs Month Heatmap (Seasonality per Shape)
shape_month = pd.crosstab(df_clean["shape"], df_clean["month"])

plt.figure(figsize=(12,6))
sns.heatmap(shape_month, cmap="magma")
plt.title("Seasonality of UFO Shapes")
plt.show()
```




```
# NLP Sentiment
from textblob import TextBlob

df_clean["sentiment"] = df_clean["summary"].fillna("").apply(lambda x: TextBlob(x).sentiment

plt.figure(figsize=(10,4))
sns.histplot(df_clean["sentiment"], bins=40)
plt.title("Sentiment of UFO Reports")
plt.show()
```



```
# ============================================
# Imports
# ============================================
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

# If you are in Jupyter and want inline plots:
# %matplotlib inline

# ============================================
# Filter to U.S. sightings (using df_clean)
# ============================================

# Make a working copy
df_us = df_clean.copy()

# Clean country column
df_us["country"] = df_us["country"].astype(str).str.strip().str.upper()

# Patterns that indicate United States in this messy dataset
us_patterns = [
    "USA",
    "U.S.A",
    "UNITED STATES",
    "UNTIED STATES",
    "USAV",
    "USAUSA",
    "USA/CANADIAN WATERS",
    "USA/MEXICO",
    "BAHAMAS/USA",
    "US AND CANADA BORDER"
]

df_us = df_us[
    df_us["country"].apply(
        lambda x: any(p in x for p in us_patterns)
    )
]

print("Rows after U.S. filtering:", df_us.shape[0])
```

```
Rows after U.S. filtering: 107574
```

```
print("Unique country values after filtering:", df_us["country"].unique())
```

```
Unique country values after filtering: ['USA' 'UNITED STATES' 'USA/CANADIAN WATERS' 'USA/MEXI
 'USAV' 'USAUSA' 'US AND CANADA BORDER' 'U.S.A.'
 'UNTIED STATES OF AMERICA']
```

```
# ============================================
# Clean and restrict to valid U.S. states
# ============================================

# Clean state codes
df_us["state"] = df_us["state"].astype(str).str.upper().str.strip()

# List of valid USPS state abbreviations
valid_states = [
    "AL","AK","AZ","AR","CA","CO","CT","DE","FL","GA",
    "HI","ID","IL","IN","IA","KS","KY","LA","ME","MD",
    "MA","MI","MN","MS","MO","MT","NE","NV","NH","NJ",
    "NM","NY","NC","ND","OH","OK","OR","PA","RI","SC",
    "SD","TN","TX","UT","VT","VA","WA","WV","WI","WY"
]

df_us = df_us[df_us["state"].isin(valid_states)]

print("Rows after restricting to valid states:", df_us.shape[0])
```

```
Rows after restricting to valid states: 107453
```

```
print("Example state counts:")
```

```
Example state counts:
```

```
print(df_us["state"].value_counts().head())
```

```
state
CA    13567
FL     6742
```

```
WA     5706
TX     4986
NY     4837
Name: count, dtype: int64
```

```
# ================================================
# Create state_counts = number of sightings per state
# ================================================

state_counts = (
    df_us.groupby("state")
         .size()
         .reset_index(name="sightings")
)

print("\nState counts (first few rows):")
```

```
State counts (first few rows):
```

```
print(state_counts.head())
```

```
  state  sightings
0    AK        539
1    AL       1114
2    AR        960
3    AZ       4096
4    CA      13567
```

```
print("Number of states in state_counts:", state_counts.shape[0])
```

```
Number of states in state_counts: 50
```

```
# ================================================
# Attach political party to each state
# (Based on 2020 presidential winner)
# ================================================

state_party = {
    "AL": "Rep", "AK": "Rep", "AZ": "Dem", "AR": "Rep", "CA": "Dem",
```

```
    "CO": "Dem", "CT": "Dem", "DE": "Dem", "FL": "Rep", "GA": "Dem",
    "HI": "Dem", "ID": "Rep", "IL": "Dem", "IN": "Rep", "IA": "Rep",
    "KS": "Rep", "KY": "Rep", "LA": "Rep", "ME": "Dem", "MD": "Dem",
    "MA": "Dem", "MI": "Dem", "MN": "Dem", "MS": "Rep", "MO": "Rep",
    "MT": "Rep", "NE": "Rep", "NV": "Dem", "NH": "Dem", "NJ": "Dem",
    "NM": "Dem", "NY": "Dem", "NC": "Rep", "ND": "Rep", "OH": "Rep",
    "OK": "Rep", "OR": "Dem", "PA": "Dem", "RI": "Dem", "SC": "Rep",
    "SD": "Rep", "TN": "Rep", "TX": "Rep", "UT": "Rep", "VT": "Dem",
    "VA": "Dem", "WA": "Dem", "WV": "Rep", "WI": "Dem", "WY": "Rep"
}

state_counts["party"] = state_counts["state"].map(state_party)

print("\nStates with missing party mapping (should be empty):")
```

```
States with missing party mapping (should be empty):
```

```
print(state_counts[state_counts["party"].isna()])
```

```
Empty DataFrame
Columns: [state, sightings, party]
Index: []
```

```
# Drop any rows without a party label (just in case)
state_counts = state_counts.dropna(subset=["party"])

print("\nState counts with party attached (first few rows):")
```

```
State counts with party attached (first few rows):
```

```
print(state_counts.head())
```

```
  state  sightings party
0    AK         539   Rep
1    AL        1114   Rep
2    AR         960   Rep
3    AZ        4096   Dem
4    CA       13567   Dem
```

```
print("Number of Dem states:", (state_counts["party"] == "Dem").sum())
```

```
Number of Dem states: 25
```

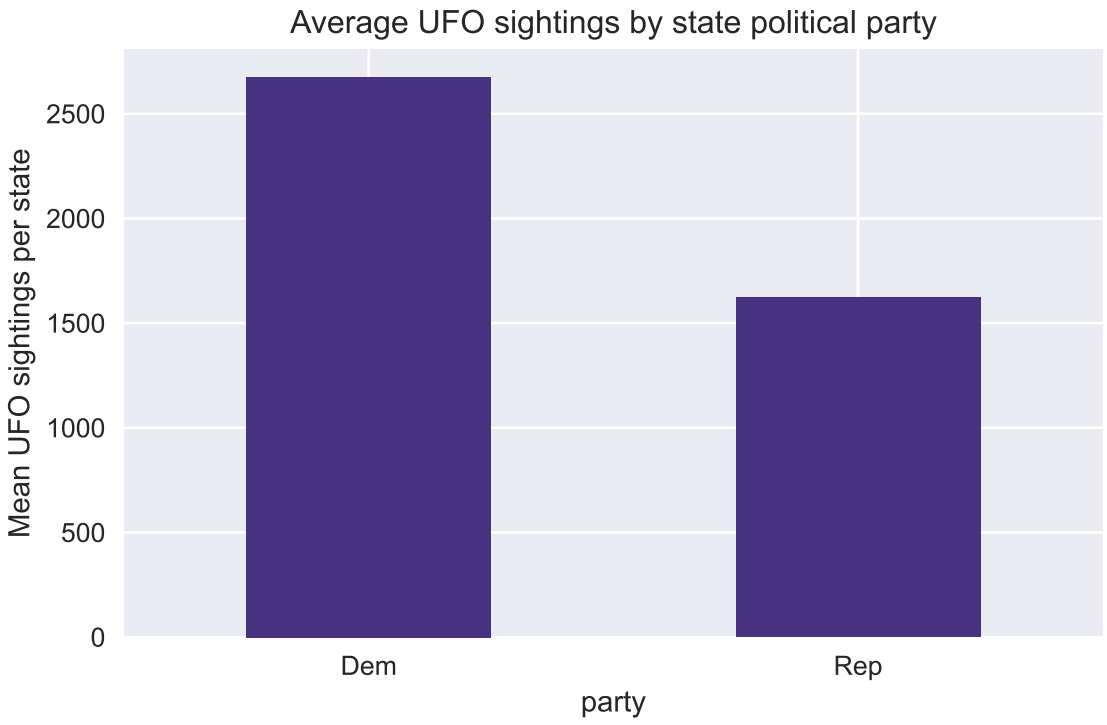
```
print("Number of Rep states:", (state_counts["party"] == "Rep").sum())
```

```
Number of Rep states: 25
```

```
# ================================================
# Visualization: bar plot of mean sightings by party
# ================================================

plt.figure(figsize=(6, 4))
(
    state_counts
    .groupby("party")["sightings"]
    .mean()
    .reindex(["Dem", "Rep"])
    .plot(kind="bar")
)
plt.ylabel("Mean UFO sightings per state")
plt.title("Average UFO sightings by state political party")
plt.xticks(rotation=0)
```

```
(array([0, 1]), [Text(0, 0, 'Dem'), Text(1, 0, 'Rep')])
```

```
plt.tight_layout()
plt.show()
```




```
# ================================================
# Hypothesis test 1:
# Two-sample t test for mean sightings per state
# ================================================

# H0: mean sightings in Dem states = mean sightings in Rep states
# H1: they are different

dem_sightings = state_counts.loc[state_counts["party"] == "Dem", "sightings"]
rep_sightings = state_counts.loc[state_counts["party"] == "Rep", "sightings"]

print("\nMean sightings per state:")
```

```
Mean sightings per state:
```

```
print("Dem states:", dem_sightings.mean())
```

```
Dem states: 2675.68
```

```
print("Rep states:", rep_sightings.mean())
```

```
Rep states: 1622.44
```

```
t_stat, p_val = stats.ttest_ind(
    dem_sightings,
    rep_sightings,
    equal_var=False   # Welch t test
)

print("\n=== Two-sample t test (Welch) ===")
```

```
=== Two-sample t test (Welch) ===
```

```
print("t statistic:", round(t_stat, 3))
```

```
t statistic: 1.712
```

```
print("p value:", p_val)
```

```
p value: 0.09492753791672308
```

```
# ================================================
# Hypothesis test 2:
# Chi square: Party vs High/Low sightings
# ================================================

# Create high vs low based on median number of sightings per state
median_count = state_counts["sightings"].median()
state_counts["sight_level"] = np.where(
    state_counts["sightings"] > median_count,
    "High",
    "Low"
)

contingency = pd.crosstab(state_counts["party"], state_counts["sight_level"])
print("\nContingency table (party x sight level):")
```

Contingency table (party x sight level):

```
print(contingency)
```

```
sight_level  High  Low
party
Dem            17    8
Rep             8   17
```

```
chi2_stat, chi2_p, dof, expected = stats.chi2_contingency(contingency)

print("\n=== Chi square test of independence ===")
```

```
=== Chi square test of independence ===
```

```
print("Chi square statistic:", round(chi2_stat, 3))
```

```
Chi square statistic: 5.12
```

```
print("Degrees of freedom:", dof)
```

```
Degrees of freedom: 1
```

```
print("p value:", chi2_p)
```

```
p value: 0.023651616655356
```

```
# ============================================
# Hypothesis test 3:
# Mann-Whitney U test: Nonparametric Alternative to t-test
# ============================================

from scipy.stats import mannwhitneyu

u_stat, p_mw = mannwhitneyu(
    dem_sightings,
```

```
    rep_sightings,
    alternative="two-sided"
)

print("Mann-Whitney U statistic:", u_stat)
```

```
Mann-Whitney U statistic: 422.0
```

```
print("p-value:", p_mw)
```

```
p-value: 0.03443756329229049
```

```
# ================================================
# Hypothesis test 4:
# Logistic regression: high/low sighting probability
# ================================================

import statsmodels.formula.api as smf
import numpy as np # Import numpy for np.where

# Create a binary variable for party (Dem=1, Rep=0) for logistic regression
state_counts['party_binary'] = state_counts['party'].apply(lambda x: 1 if x == 'Dem' else 0)

# Create a binary variable for high sighting (High=1, Low=0)
state_counts['is_high_sighting'] = np.where(state_counts['sight_level'] == 'High', 1, 0)

logit_model = smf.logit(
    formula="is_high_sighting ~ party_binary",
    data=state_counts
).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.626869
         Iterations 5
```

```
print(logit_model.summary())
```

```
                           Logit Regression Results
==============================================================================
```

```
Dep. Variable:       is_high_sighting   No. Observations:                   50
Model:                         Logit   Df Residuals:                       48
Method:                          MLE   Df Model:                            1
Date:                Fri, 12 Dec 2025   Pseudo R-squ.:                0.09562
Time:                        17:46:59   Log-Likelihood:               -31.343
converged:                       True   LL-Null:                      -34.657
Covariance Type:            nonrobust   LLR p-value:                  0.01004
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      -0.7538      0.429     -1.758      0.079      -1.594       0.087
party_binary    1.5075      0.606      2.486      0.013       0.319       2.696
==============================================================================
```

```python
# ====================================================
# Hypothesis test 5:
# Poisson regression
# ====================================================

import statsmodels.api as sm
import statsmodels.formula.api as smf

state_counts["party_binary"] = (state_counts["party"] == "Dem").astype(int)

poisson_model = smf.glm(
    formula="sightings ~ party_binary",
    data=state_counts,
    family=sm.families.Poisson()
).fit()

print(poisson_model.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:              sightings   No. Observations:                   50
Model:                            GLM   Df Residuals:                       48
Model Family:                 Poisson   Df Model:                            1
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -36372.
Date:                Fri, 12 Dec 2025   Deviance:                       72288.
Time:                        17:47:00   Pearson chi2:                 9.86e+04
No. Iterations:                     5   Pseudo R-squ. (CS):              1.000
```

```
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        7.3917      0.005   1488.668      0.000       7.382       7.401
party_binary     0.5003      0.006     79.495      0.000       0.488       0.513
==============================================================================
```

```
# ========================================================
# Hypothesis test 6:
# Effect sizes (Cohen's d, Cramér's V): show magnitude
# ========================================================

# Cohen's d (strength of mean difference)
import numpy as np

d = (dem_sightings.mean() - rep_sightings.mean()) / np.sqrt(
    (dem_sightings.var() + rep_sightings.var()) / 2
)
print("Cohen's d:", d)
```

```
Cohen's d: 0.4842789757757099
```

```
# Cramér's V (strength of association for chi-square)
import numpy as np

n = contingency.sum().sum()
phi2 = chi2_stat / n
r, k = contingency.shape
cramers_v = np.sqrt(phi2 / min(k-1, r-1))

print("Cramér's V:", cramers_v)
```

```
Cramér's V: 0.32
```

```
# ========================================================
# Hypothesis test 7:
# Permutation test: voids distributional assumptions
# ========================================================

import numpy as np
```

```
observed_diff = dem_sightings.mean() - rep_sightings.mean()

combined = np.concatenate([dem_sightings, rep_sightings])
n_dem = len(dem_sightings)

num_permutations = 5000
diffs = []

for _ in range(num_permutations):
    np.random.shuffle(combined)
    new_dem = combined[:n_dem]
    new_rep = combined[n_dem:]
    diffs.append(new_dem.mean() - new_rep.mean())

p_perm = np.mean(np.abs(diffs) >= np.abs(observed_diff))

print("Permutation test p-value:", p_perm)
```

```
Permutation test p-value: 0.0846
```

```
# ============================================================
# Word clouds by political party (using "summary" text)
# ============================================================

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

# Optional: add custom stopwords to remove very common words
custom_stopwords = set(STOPWORDS)
custom_stopwords.update([
    "ufo", "ufos", "object", "light", "lights", "saw", "like",
    "just", "one", "two", "three", "looked"
])

# FIX: Merge the 'party' information into df_us
# state_counts contains 'state' and 'party' columns.
# We need to merge this back into df_us based on the 'state' column.
df_us = df_us.merge(state_counts[['state', 'party']], on='state', how='left')


# Build text for each party
```

```
dem_text = " ".join(
    df_us.loc[df_us["party"] == "Dem", "summary"]
         .dropna()
         .astype(str)
)
rep_text = " ".join(
    df_us.loc[df_us["party"] == "Rep", "summary"]
         .dropna()
         .astype(str)
)

# Generate word clouds
wc_dem = WordCloud(
    width=1200,
    height=600,
    background_color="black",
    stopwords=custom_stopwords
).generate(dem_text)

wc_rep = WordCloud(
    width=1200,
    height=600,
    background_color="black",
    stopwords=custom_stopwords
).generate(rep_text)

# Plot side by side
plt.figure(figsize=(16, 7))
```

```
<Figure size 1600x700 with 0 Axes>
```

```
plt.subplot(1, 2, 1)
```

```
<Axes: >
```

```
plt.imshow(wc_dem, interpolation="bilinear")
```

```
<matplotlib.image.AxesImage object at 0x000002DBCC93C2B0>
```

```
plt.title("Democratic States: UFO Report Word Cloud")
```

```
Text(0.5, 1.0, 'Democratic States: UFO Report Word Cloud')
```

```
plt.axis("off")
```

```
(-0.5, 1199.5, 599.5, -0.5)
```

```
plt.subplot(1, 2, 2)
```

```
<Axes: >
```

```
plt.imshow(wc_rep, interpolation="bilinear")
```

```
<matplotlib.image.AxesImage object at 0x000002DBE8357070>
```

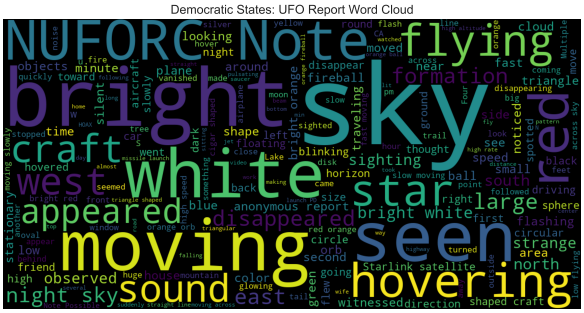
```
plt.title("Republican States: UFO Report Word Cloud")
```

```
Text(0.5, 1.0, 'Republican States: UFO Report Word Cloud')
```

```
plt.axis("off")
```

```
(-0.5, 1199.5, 599.5, -0.5)
```

```
plt.tight_layout()
plt.show()
```



```
# ============================================================
# Duration cleaning for US data and party labels
# ============================================================

# Work on a fresh copy so we do not overwrite earlier analysis
df_us_duration = df_us.copy()

# Extract numeric part from the "duration" text column
# Example: "5 minutes", "10 mins", "about 30 sec"
df_us_duration["duration_numeric"] = (
    df_us_duration["duration"]
    .astype(str)
    .str.extract(r"(\d+\.?\d*)")[0]    # first capture group
)

df_us_duration["duration_numeric"] = pd.to_numeric(
    df_us_duration["duration_numeric"],
    errors="coerce"
)

# Drop rows where we could not parse a numeric duration
df_us_duration = df_us_duration.dropna(subset=["duration_numeric"])

# Optional: remove extreme outliers (top 1 percent)
upper_cut = df_us_duration["duration_numeric"].quantile(0.99)
df_us_duration = df_us_duration[df_us_duration["duration_numeric"] <= upper_cut]

print("Number of rows with usable duration:", df_us_duration.shape[0])
```

```
Number of rows with usable duration: 106492
```

```
# ============================================================
# Average duration by political party
# ============================================================

duration_by_party = (
    df_us_duration
    .groupby("party")["duration_numeric"]
    .agg(["mean", "median", "std", "count"])
    .reset_index()
)
```

```
print("Duration summary by party:")
```

```
Duration summary by party:
```

```
print(duration_by_party)
```

```
  party       mean  median        std  count
0   Dem  10.554700     5.0  11.615218  66287
1   Rep  10.702968     5.0  11.737357  40205
```

```
# Bar plot of mean duration by party
plt.figure(figsize=(6, 4))
```

```
<Figure size 600x400 with 0 Axes>
```

```
plt.bar(
    duration_by_party["party"],
    duration_by_party["mean"]
)
```

```
<BarContainer object of 2 artists>
```

```
plt.xlabel("Political party")
```

```
Text(0.5, 0, 'Political party')
```

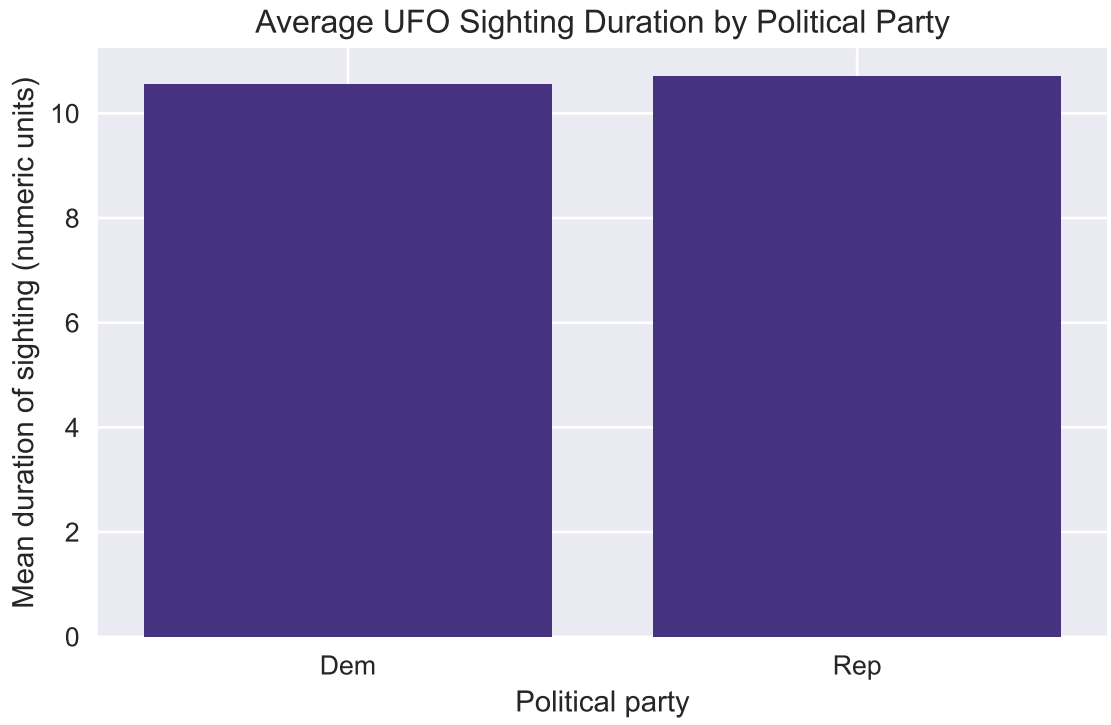
```
plt.ylabel("Mean duration of sighting (numeric units)")
```

```
Text(0, 0.5, 'Mean duration of sighting (numeric units)')
```

```
plt.title("Average UFO Sighting Duration by Political Party")
```

```
Text(0.5, 1.0, 'Average UFO Sighting Duration by Political Party')
```

```
plt.tight_layout()
plt.show()
```

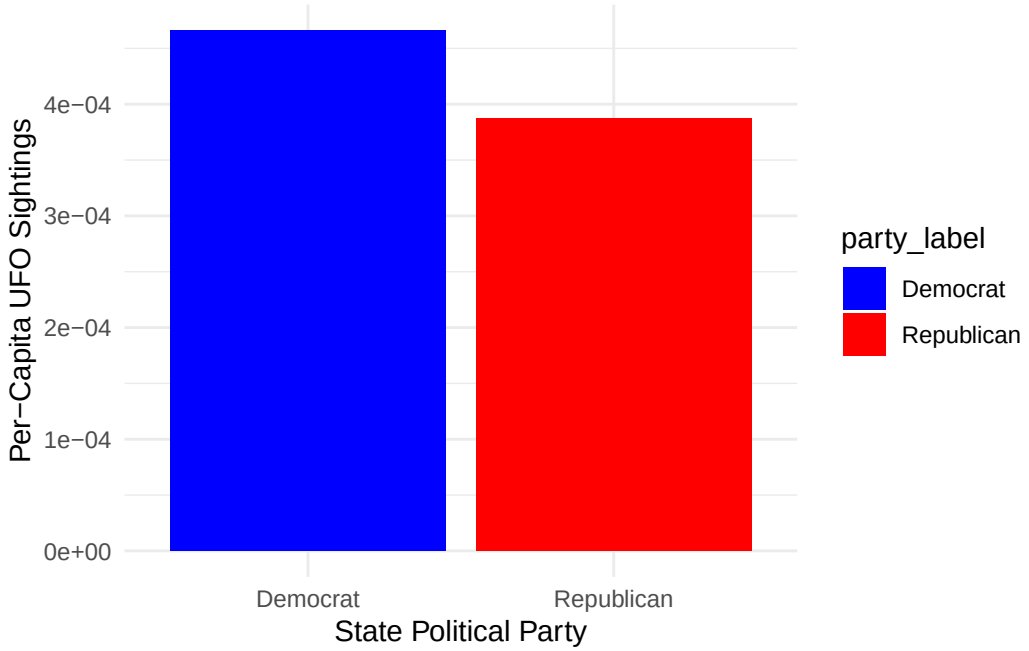



```
library(dplyr)
library(ggplot2)

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

sightings <- c(
  577, 1216, 1048, 4384, 14502, 2820, 1704, 374, 7263, 2395,
  614, 1100, 1176, 3777, 2166, 1042, 1497, 1001, 2394, 1623,
  1069, 3238, 1872, 2430, 691, 863, 3301, 226, 626, 1029,
  2538, 1451, 1501, 5224, 3878, 1338, 3151, 4442, 559, 2010,
  340, 2028, 5442, 1359, 2362, 554, 6126, 2232, 801, 366)
```

```r
population <- c(
  733391, 5024279, 3011524, 7151502, 39538223, 5773714, 3605944, 989948, 21538187, 10711908,
  1455271, 3190369, 1839106, 12812508, 6785528, 2937880, 4505836, 4661468, 7029917, 6177224,
  1362359, 10077331, 5706494, 6154913, 2961279, 1084225, 10439388, 779094, 1961504, 1377529,
  9288994, 2117522, 3104614, 20201249, 11799448, 3959353, 4237256, 13002700, 1097379, 5118425
  886667, 6910840, 29145505, 3271616, 8631393, 643077, 7705281, 5893718, 1793716, 576851)

#0 for republican, 1 for democrat
party <- c(
  0,0,0,1,1,1,1,1,0,1,
  1,0,0,1,0,0,0,0,1,1,
  1,1,1,0,0,0,0,0,0,1,
  1,1,1,1,0,0,1,1,1,0,
  0,0,0,0,1,1,1,1,0,0)

data <- data.frame(states, sightings, population, party) %>%
  mutate(capita = sightings / population)

wilcox.test(capita ~ party, data = data)
```

```
    Wilcoxon rank sum exact test

data:  capita by party
W = 243, p-value = 0.1823
alternative hypothesis: true location shift is not equal to 0
```

```r
plotdata <- data %>% group_by(party) %>%
  summarize(meancapita = mean(capita)) %>%
  mutate(party_label = ifelse(party == 1, "Democrat", "Republican"))

ggplot(plotdata, aes(x = party_label, y = meancapita, fill = party_label)) +
  geom_bar(stat = "identity") +
  labs(x = "State Political Party", y = "Per-Capita UFO Sightings") +
  scale_fill_manual(values = c("Democrat" = "blue", "Republican" = "red")) +
  theme_minimal()
```



References

French, C. C., J. Santomauro, V. Hamilton, R. Fox, and M. A. Thalbourne. 2008. "Psychological Aspects of the Alien Contact Experience." https://www.sciencedirect.com/science/article/pii/S0010945208001408.

Medina, R. M., S. C. Brewer, and S. M. Kirkpatrick. 2023. "An Environmental Analysis of Public UAP Sightings and Sky View Potential." https://pmc.ncbi.nlm.nih.gov/articles/PMC10721628.

Renner, T. 2020. "UFO Sightings 1969 to 2019." 2020. https://www.kaggle.com/datasets/fireballbyedimyrnmom/ufo-sightings-1969-to-2019.

Stephan, K. D., S. Ghimire, W. A. Stapleton, and J. Bunnell. 2009. "Spectroscopy Applied to Observations of Terrestrial Light Sources." https://pubs.aip.org/aapt/ajp/article/77/8/697/310879/Spectroscopy-applied-to-observations-of.

U.S. Census Bureau. 2021. "2020 Census Apportionment Results, Table 2." 2021. https://www2.census.gov/programs-surveys/decennial/2020/data/apportionment/apportionment-2020-table02.pdf.

Wikimedia Foundation. 2025. "Red States and Blue States." 2025. https://en.wikipedia.org/wiki/Red_states_and_blue_states.