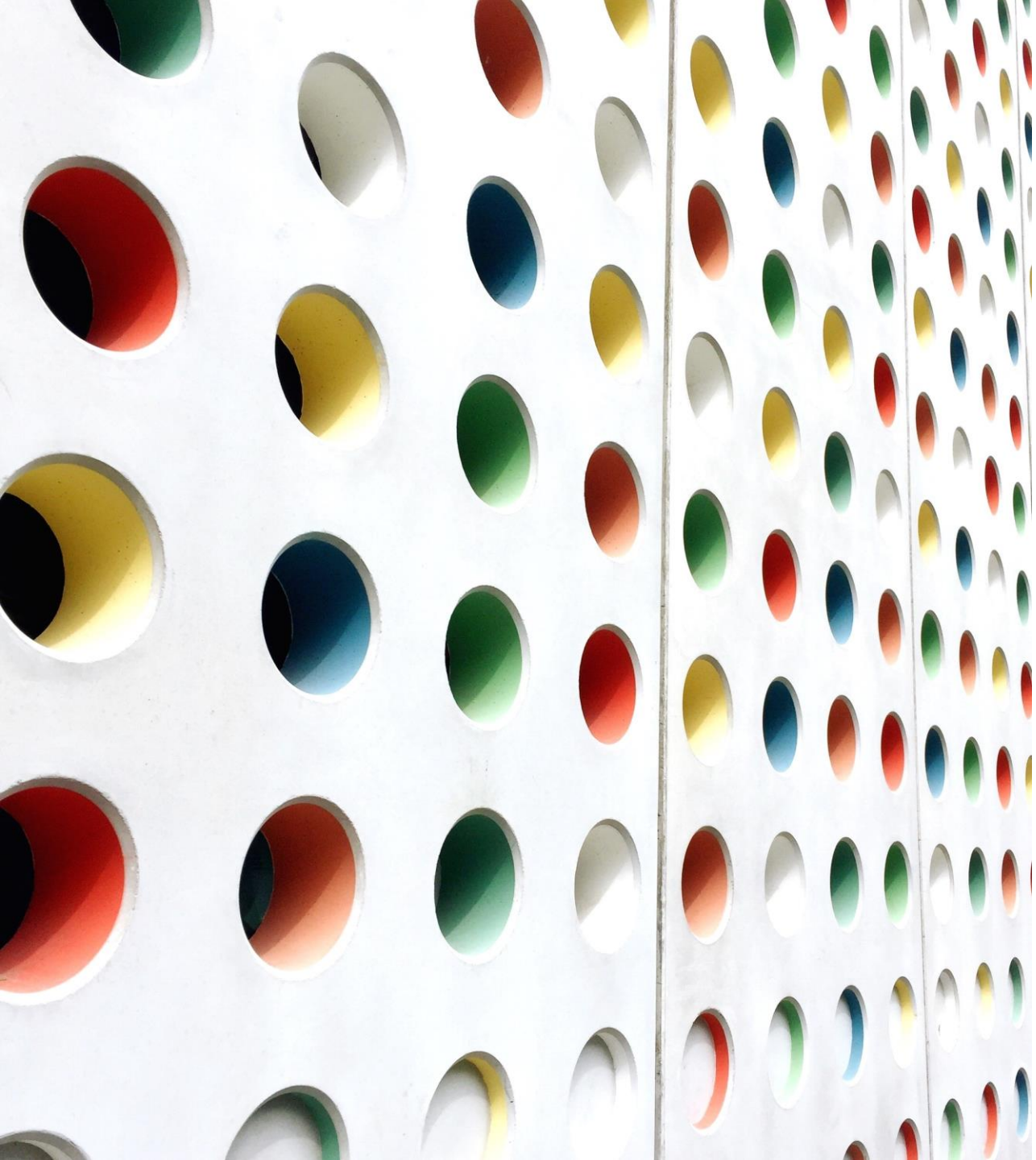


UFO Sightings Analysis

ANTONIO PAUL MAMMONE

Data Frame & Cleaning

- I Decided on the UFO sitings DF from: https://www.kaggle.com/datasets/NUFORC/ufo-sightings

- Has over 80k rows

- .dt.year - Extract year as integer

- .dt.month - Extract month (1-12)

- .dt.day - Extract day of month

- .dt.hour - Extract hour (0-23)

- .dt.day_name() - Get day name as string ('Monday', 'Tuesday', etc.)

- .dt.dayofweek - Get day as number (0=Monday, 6=Sunday)

- pd.to_datetime() - Convert string dates to datetime objects

- errors='coerce' - Gracefully handle invalid dates (converts to NaT)

- pd.to_numeric() - Convert duration strings to numbers

- errors='coerce' - Handle non-numeric values gracefully

- .dropna(subset=['datetime', 'city', 'state']) - Remove rows with missing critical data

- Boolean filtering: (df_clean['year'] >= 1950) & (df_clean['year'] <= 2023)

datetime	city	state	country	shape	duration (	duration (	comment:	date poste	latitude	longitude
########	san marco	tx	us	cylinder	2700	45 minute	This even	4/27/2004	29.88306	-97.9411
########	lackland a	tx		light	7200	1-2 hrs	1949 Lackl	12/16/200	29.38421	-98.5811
########	chester (uk/england		gb	circle	20	20 second	Green/Or	1/21/2008	53.2	-2.91667
########	edna	tx	us	circle	20	1/2 hour	My older	1/17/2004	28.97833	-96.6458
########	kaneohe	hi	us	light	900	15 minute	AS a Marir	1/22/2004	21.41806	-157.804
########	bristol	tn	us	sphere	300	5 minutes	My father	4/27/2007	36.595	-82.1889
########	penarth (uk/wales)		gb	circle	180	about 3 m	penarth u	2/14/2006	51.43472	-3.18
########	norwalk	ct	us	disk	1200	20 minute	A bright o	########	41.1175	-73.4083
########	pell city	al	us	disk	180	3 minutes	Strobe Lig	3/19/2009	33.58611	-86.2861
########	live oak	fl	us	disk	120	several m	Saucer zag	########	30.29472	-82.9842
########	hawthorn	ca	us	circle	300	5 min.	ROUND &	10/31/200	33.91639	-118.352
########	brevard	nc	us	fireball	180	3 minutes	silent red	########	35.23333	-82.7344
########	bellmore	ny	us	disk	1800	30 min.	silver disc	########	40.66861	-73.5275
########	manchest	ky	us	unknown	180	3 minutes	Slow mov	2/14/2008	37.15361	-83.7619
########	lexington	nc	us	oval	30	30 second	green ova	2/14/2010	35.82389	-80.2536
########	harlan cou	ky	us	circle	1200	20minute:	On octobe	9/15/2005	36.84306	-83.3219
########	west bloo	mi	us	disk	120	2 minutes	The UFO v	8/14/2007	42.53778	-83.2331
########	niantic	ct	us	disk	1800	20-30 min	Oh&#44 w	9/24/2003	41.32528	-72.1936
########	bermuda	nas		light	20	20 sec.	saw fast n	########	32.36417	-64.6786

```
# Data Cleaning and Preparation
# Convert datetime columns
df['datetime'] = pd.to_datetime(df['datetime'], errors='coerce')
df['date posted'] = pd.to_datetime(df['date posted'], errors='coerce')

# Extract time features for analysis
df['year'] = df['datetime'].dt.year
df['month'] = df['datetime'].dt.month
df['day'] = df['datetime'].dt.day
df['hour'] = df['datetime'].dt.hour
df['dayofweek'] = df['datetime'].dt.day_name()
df['dayofweek_num'] = df['datetime'].dt.dayofweek
df['is_weekend'] = df['dayofweek'].isin(['Friday', 'Saturday', 'Sunday'])

# Clean duration column - convert to numeric
df['duration_seconds'] = pd.to_numeric(df['duration (seconds)'], errors='coerce')

# Remove rows with missing critical data
df_clean = df.dropna(subset=['datetime', 'city', 'state'])

# Filter for reasonable years (1950-2023)
df_clean = df_clean[(df_clean['year'] >= 1950) & (df_clean['year'] <= 2023)]

print(f"Original dataset: {len(df)} rows")
print(f"Cleaned dataset: {len(df_clean)} rows")
print(f"Removed: {len(df) - len(df_clean)} rows ({((len(df) - len(df_clean))/len(df)*100):.2f}%)")
```

```
Original dataset: 88875 rows
Cleaned dataset: 80152 rows
Removed: 8723 rows (9.81%)
```

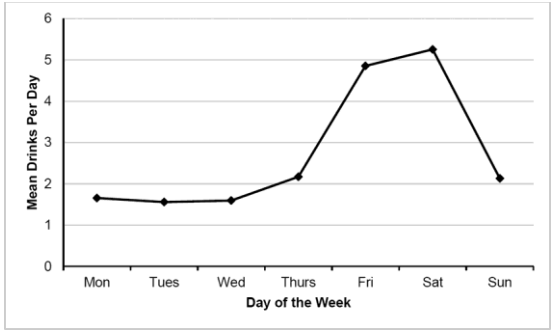
Represent dates hidden until clicked on ☺

UFO Sightings by Day of Week

- First, I created a day_name column using df['datetime'].dt.day_name()

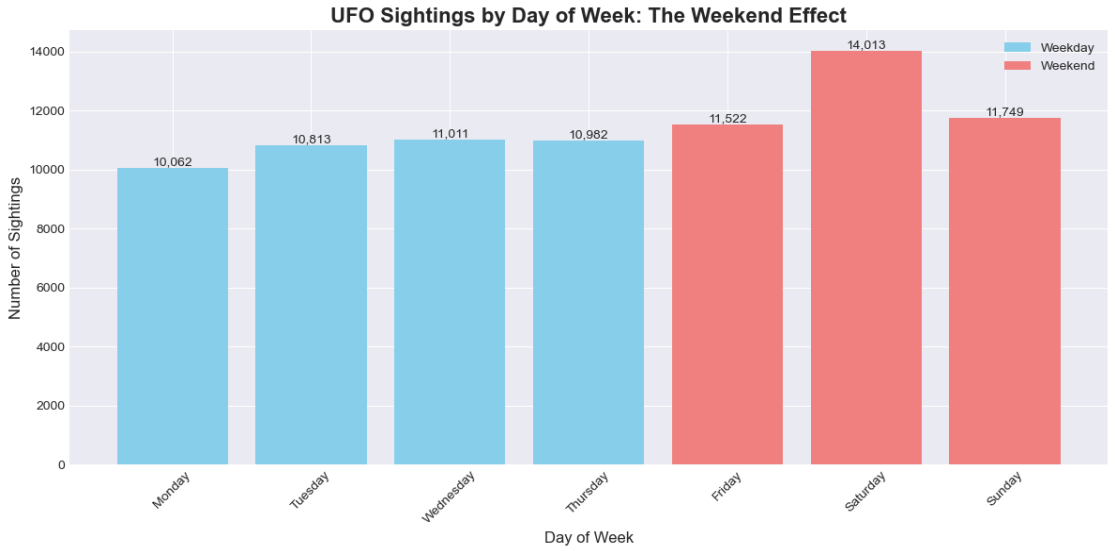
- Then I used value_counts() with reindex to ensure days appear in order

- I applied different colors using a list comprehension to distinguish weekend



Mean drinks of alcohol consumed by day of the week:
https://www.researchgate.net/figure/Mean-drinks-of-alcohol-consumed-by-day-of-the-week_fig1_304069436

```
Weekend Analysis:
Total weekend sightings: 37,284
Total weekday sightings: 42,868
Average per weekend day: 12,428
Average per weekday: 10,717
Weekend boost: 16.0%
```




```python
# Plot 1 - The Weekend Effect (Bar Plot)

plt.figure(figsize=(12, 6))

# Order days properly
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
day_counts = df_clean['dayofweek'].value_counts().reindex(day_order)

# Create bar plot with different colors for weekends
colors = ['skyblue', 'skyblue', 'skyblue', 'skyblue', 'lightcoral', 'lightcoral', 'lightcoral']
bars = plt.bar(day_order, day_counts.values, color=colors)

# Add value labels on bars
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height,
             f'{int(height):,}',
             ha='center', va='bottom')

plt.title('UFO Sightings by Day of Week: The Weekend Effect', fontsize=16, fontweight='bold')
plt.xlabel('Day of Week', fontsize=12)
plt.ylabel('Number of Sightings', fontsize=12)
plt.xticks(rotation=45)

# Add legend
from matplotlib.patches import Patch
legend_elements = [Patch(facecolor='skyblue', label='Weekday'),
                   Patch(facecolor='lightcoral', label='Weekend')]
plt.legend(handles=legend_elements, loc='upper right')

plt.tight_layout()
plt.show()

# Calculate weekend vs weekday statistics
weekend_sightings = df_clean[df_clean['is_weekend']].shape[0]
weekday_sightings = df_clean[~df_clean['is_weekend']].shape[0]
weekend_avg = weekend_sightings / 3  # 3 weekend days
weekday_avg = weekday_sightings / 4  # 4 weekdays

print(f"\nWeekend Analysis:")
print(f"Total weekend sightings: {weekend_sightings:,}")
print(f"Total weekday sightings: {weekday_sightings:,}")
print(f"Average per weekend day: {weekend_avg:,.0f}")
print(f"Average per weekday: {weekday_avg:,.0f}")
print(f"Weekend boost: {((weekend_avg - weekday_avg) / weekday_avg * 100):.1f}%")
```

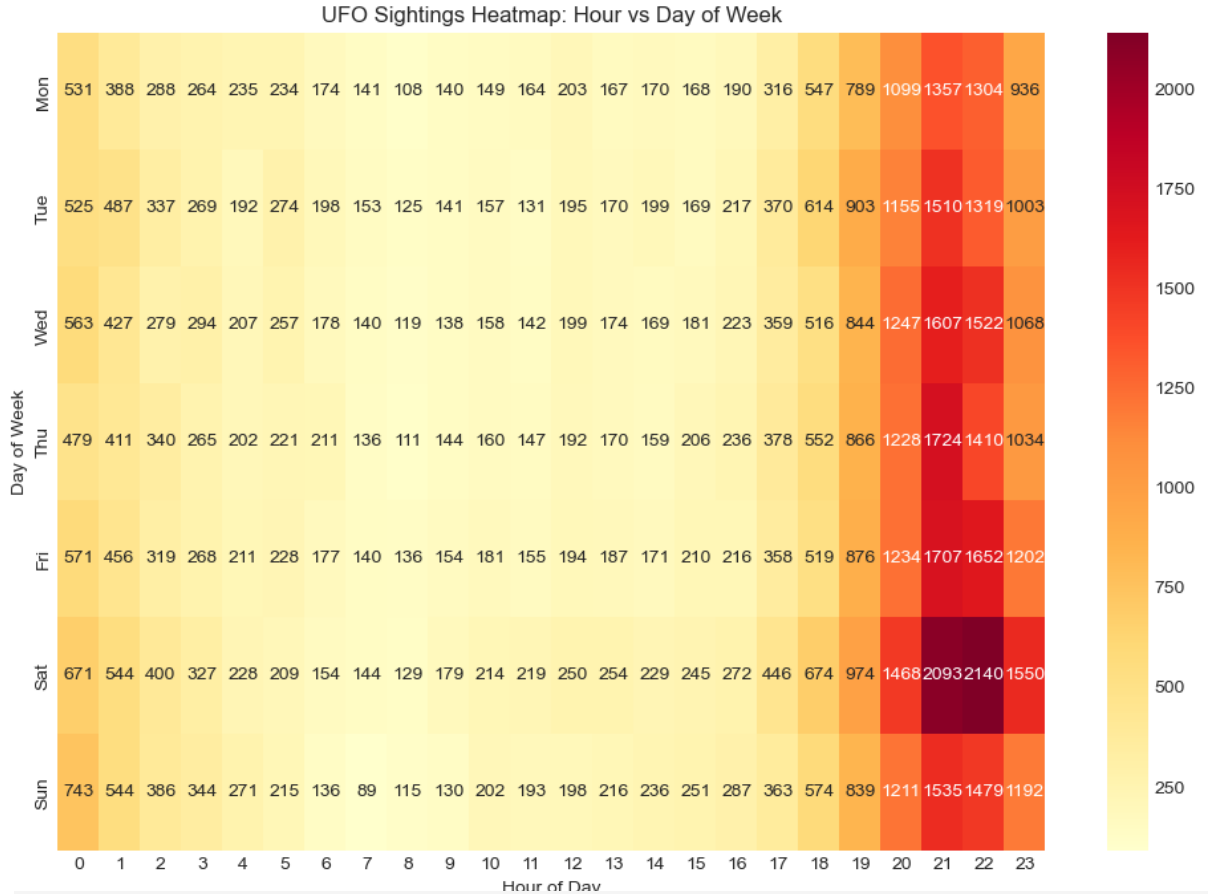
UFO Sightings Heatmap

- I used pivot_table to reshape the data: 7 df.pivot_table(values='city', index='hour', columns='day_name', aggfunc='count') 8

- This counts sightings for each hour-day combination

- I used seaborn's heatmap with the YlOrRd colormap for clear intensity visualization

- The annot=True parameter adds count values to each cell




```
# Plot 8 - Sightings by Hour and Day of Week (Heatmap)
hourly_dow = df_clean.groupby(['dayofweek_num', 'hour']).size().unstack(fill_value=0)

plt.figure(figsize=(12, 8))
sns.heatmap(hourly_dow, cmap='YlOrRd', annot=True, fmt='d',
            xticklabels=range(24),
            yticklabels=['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])
plt.title('UFO Sightings Heatmap: Hour vs Day of Week')
plt.xlabel('Hour of Day')
plt.ylabel('Day of Week')
plt.show()
```

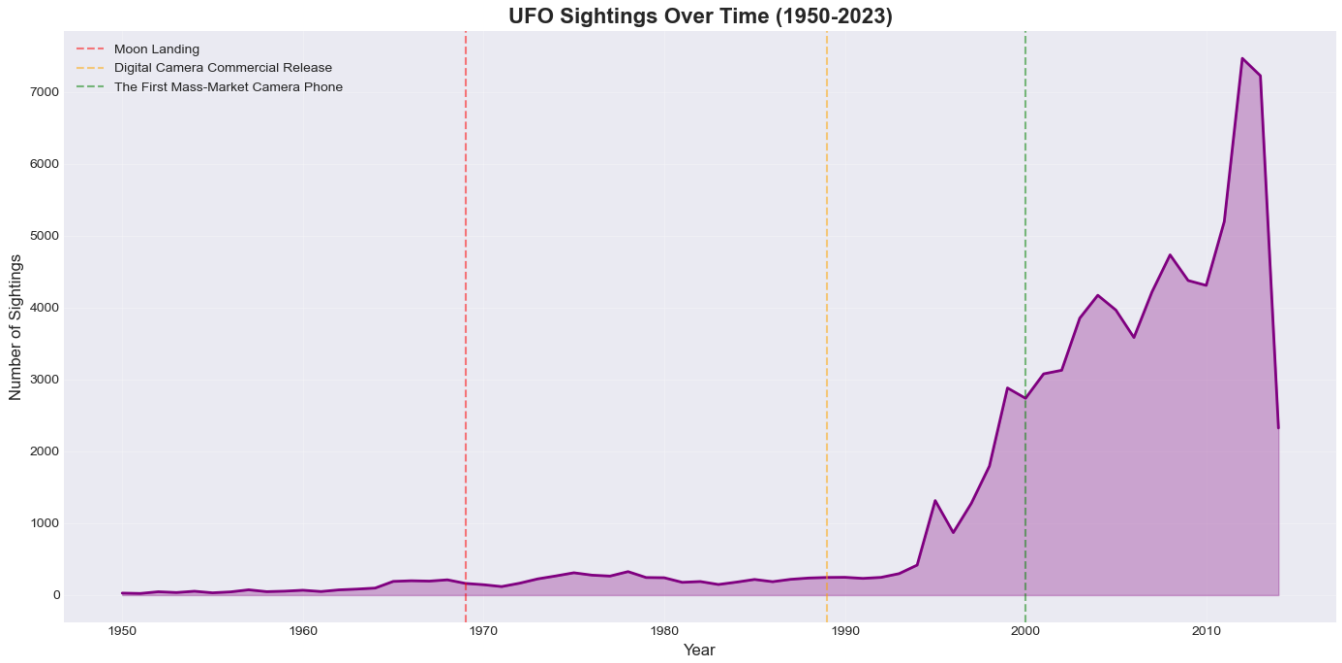
UFO Sightings Over Time

- Grouped the data by year using df.groupby('year').size() 7

- Used matplotlib's plot function with fill_between for visual impact

- Added vertical lines to mark significant historical events

```
Peak UFO Sighting Years:
2012.0: 7,470 sightings
2013.0: 7,228 sightings
2011.0: 5,199 sightings
2008.0: 4,735 sightings
2009.0: 4,378 sightings
```




```python
# Plot 3 - Sightings Over Time (Line Plot with Date Splitting)

# Group by year and count sightings
yearly_sightings = df_clean.groupby('year').size()

plt.figure(figsize=(14, 7))
plt.plot(yearly_sightings.index, yearly_sightings.values, linewidth=2, color='purple')
plt.fill_between(yearly_sightings.index, yearly_sightings.values, alpha=0.3, color='purple')

# Mark significant years
plt.axvline(x=1969, color='red', linestyle='--', alpha=0.5, label='Moon Landing')
plt.axvline(x=1989, color='orange', linestyle='--', alpha=0.5, label='Digital Camera Commercial Release')
plt.axvline(x=2000, color='green', linestyle='--', alpha=0.5, label='The First Mass-Market Camera Phone')

plt.title('UFO Sightings Over Time (1950-2023)', fontsize=16, fontweight='bold')
plt.xlabel('Year', fontsize=12)
plt.ylabel('Number of Sightings', fontsize=12)
plt.legend()
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

# Find peak years
peak_years = yearly_sightings.nlargest(5)
print("Peak UFO Sighting Years:")
for year, count in peak_years.items():
    print(f"{year}: {count:,} sightings")
```

Final Comments & Other Findings

```
=== UFO SIGHTINGS ANALYSIS SUMMARY ===

1. THE WEEKEND EFFECT:
   - 46.5% of all UFO sightings occur on weekends (Fri-Sun)
   - Weekend sightings are higher than weekdays
   - Late night weekend sightings are 0.93x more common

2. GEOGRAPHIC HOTSPOTS:
   - Top state: CA with 9,461 sightings
   - US accounts for 86.6% of all sightings

3. TEMPORAL PATTERNS:
   - Peak year: 2012.0 with 7,470 sightings
   - Peak month: Jul with 9,492 sightings
   - Peak hour: 21:00

4. UFO CHARACTERISTICS:
   - Most common shape: light

5. INTERESTING FINDINGS:
   - Summer months show significantly more sightings
   - Sightings dramatically increased after 1990s
   - Coastal states dominate the top sighting locations
   - The 'weekend effect' is real - supporting our hypothes
```