This research report shows the methodology and conclusions of finding some prevailing attributes of UFO (undefined flying object) by analyzing the words frequencies from 134k descriptions of UFO eyewitnesses.

What are the prevailing attributes of UFO

Project 1 Final Written Report

The University of Adelaide
Computer Science (Advanced)
Jiajun Yu
a1806320

Project 1 Final Written Report

Topic Introduction

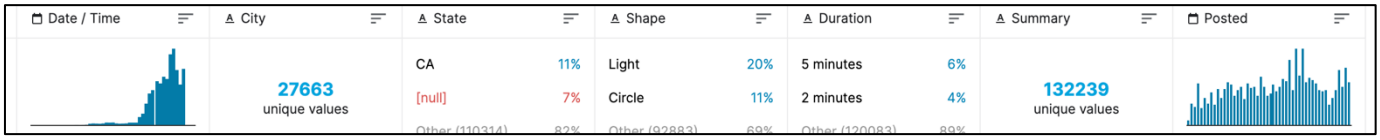
In 1977, Peter Sturrock, a professor of space science and astrophysics at Stanford University, did a research and found that 4.6% of members of the American Astronomical Society claimed that they have witnessed inexplicable aerial phenomena. This rate is close to the fact that approximately 5 percent of UFO (undefined flying object) sightings that are never explained (Knuth 2018). The topic of UFOs attracts people from all over the world. Investigations and studies for UFO have been conducted by various governments worldwide nowadays, along with private individuals and organizations (Daniela, G et al. 2021).

Motivation

Few months ago, the US government released a report about UAP (unidentified aerial phenomena). It is based on the review of 144 significant UAP reports involving observations made by military aviators between 2004 and 2021. However, the task force could only give a concrete explanation for one phenomenon, and the rest remain unexplained (Stieb & Danner 2021). That report raised the awareness of people who are concerned with UFO events. Hence, doing a research about identifying some prevailing attributes of UFO is necessary to help people know more about this unfamiliar field. This report will show the research methodology and conclusions in detail.

Dataset

The dataset utilized for this research is based on web scraped data from USA National UFO Reporting Center (Kaggle 2021). It contains 7 unique columns with 134,421 records of UFO sightings from 1946 to 2021 in the USA and was updated in July 2021. This dataset includes information such as time, locations, and descriptions from eyewitnesses about UFO sightings.



Methodology

This research mainly analyses the words frequencies of information contained in the dataset and draws conclusions by categorizing words to identify the potential patterns of UFOs. The research has explored the shape, color, USA state positions, audio, and general positions patterns of UFO. In addition, appearance year, month, time, and duration patterns of UFO have also been investigated. All visualizations were implemented by Python. Three main methods were adopted to explore the main characteristics of UFO, which were ‘All Words Frequencies Counting’ code (Code 1), ‘Target Words Frequencies Counting’ code (Code 2) and Microsoft Excel respectively. Firstly, ‘All Words Frequencies Counting’ code is able to count the frequency for every word in all descriptions. ‘Target Words Frequencies Counting’ code can count the word frequency for several specific words. These two scripts will not count repeated words from one piece of description. Lastly, Microsoft Excel has powerful searching functions which can identify some patterns that cannot be done with Python.

To find the shape pattern of UFO, ‘All Words Frequencies Counting’ code was used to obtain the word frequency for all words in a specific column of the dataset containing the shape information of UFO. After getting the frequencies for words describing shapes, the outcome was visualized

(Figure 1). However, Figure 1 only shows 10 shapes of UFO. In order to show the overall shape pattern of UFO, word cloud visualization was utilized (Figure 2). It took a text file as its input which contained different numbers of shape words according to their frequencies and was generated by Code 15. This visualization is able to show the frequencies for all shape words by plotting them with various sizes. From the data, it can be concluded that the top 3 words describing the shapes of UFO are 'circle', 'triangle' and 'fireball', which totally account for 38.85% of all records. Taking into the consideration of people's stereotype towards UFO, the ratio of circular shapes (circle, fireball, sphere, disk, oval, egg) should be calculated. It turned out to be 57.21% which indicates that people might be affected by their stereotype towards UFO.

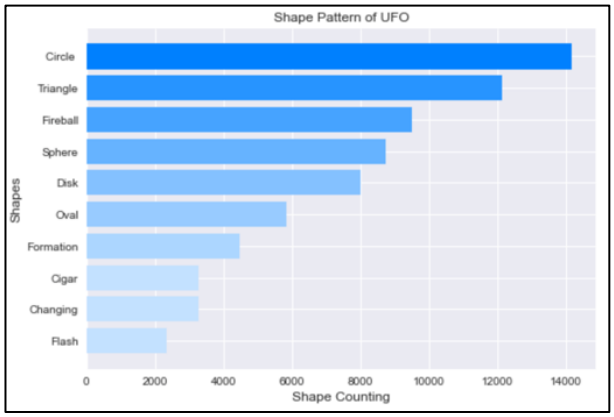


Figure 1

Generated by Code4

Figure 2

Generated by Code7

To find the color pattern of UFO, 'All Words Frequencies Counting' code was utilized to count the word frequency for all words from people's descriptions. Then, color words as well as their words frequencies were recorded and visualized (Figure 3). To make Figure 3 more intuitive, waffle visualization was utilized (Figure 4). From the data, it can be concluded that the top 3 color words are 'bright', 'white' and 'orange', which totally account for 59.44% of all records. In addition, most colors are bright colors, which means that the vast majority of people might witness UFOs at night.

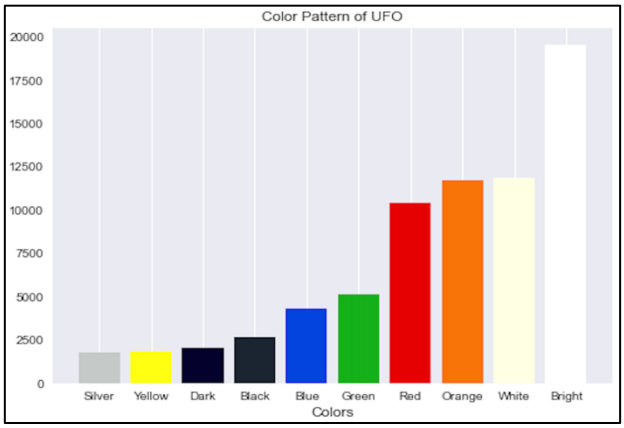


Figure 3

Generated by Code3

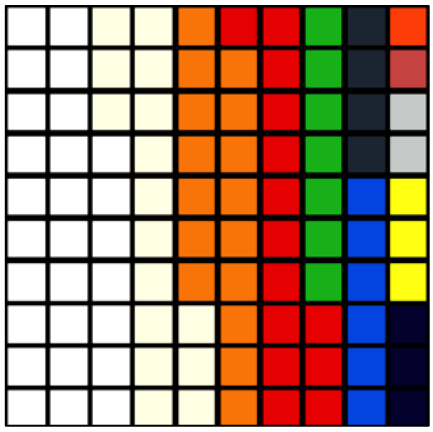


Figure 4

Generated by Code13

The method of identifying the USA state positions of UFO is identical to finding the shape pattern of UFO. Using 'All Words Frequencies Counting' code to acquire the word frequency for state names in a specific column of the dataset could produce a list of state names with the corresponding numbers of UFO witnessed in those states. After that, the outcome was visualized

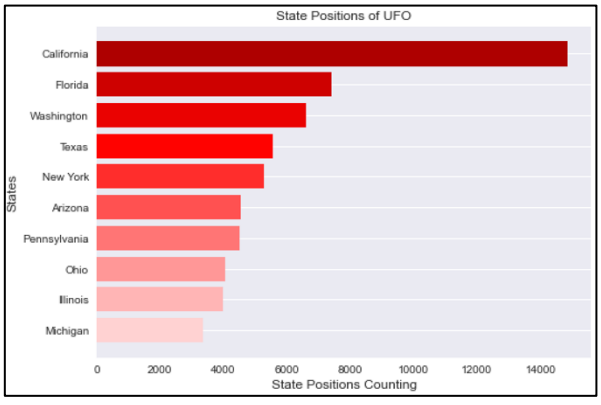


Figure 5

Generated by Code5

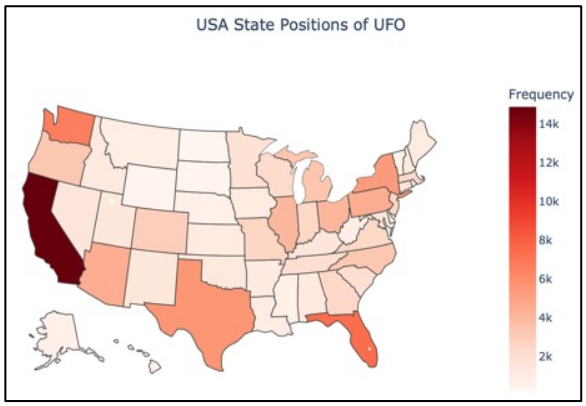


Figure 6

Generated by Code12

(Figure 5). In order to have an overview of the distribution of UFO events, map visualization was adopted (Figure 6). It shows the overall number of UFO events in each state that witnessed from 1946 to 2021. It can be concluded from the data that people are more likely to witness UFOs in California, Florida, and Washington, and the ratio of UFOs to be witnessed in California is 12.44%. In addition, there are 76.61% of UFOs that were witnessed in coastal states.

Finding the general positions of UFO needs both 'All Words Frequencies Counting' code and 'Target Words Frequencies Counting' code. The first script was used to find all words related to general positions, and the second script was utilized to find the word frequency for a group of similar words. It can be seen from Figure 7 that it is easier for people to see UFOs from home, on the mountain, and on the street. In addition, the percentage of witnessing UFOs at home is about 41.39% higher than on the mountain, and it accounts for approximately 19.74% of all records.

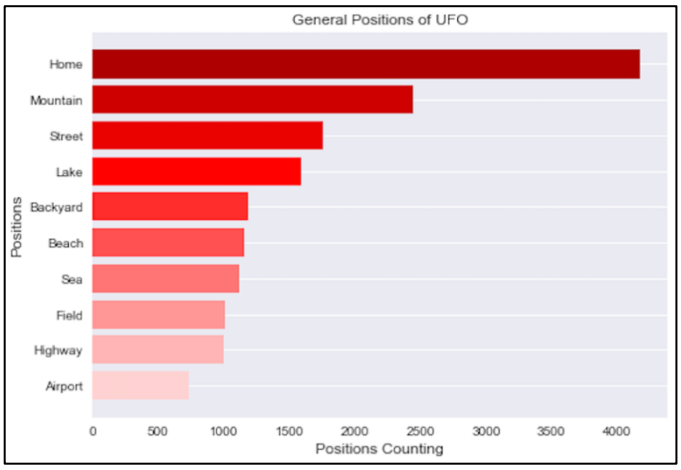


Figure 7
Generated by Code10

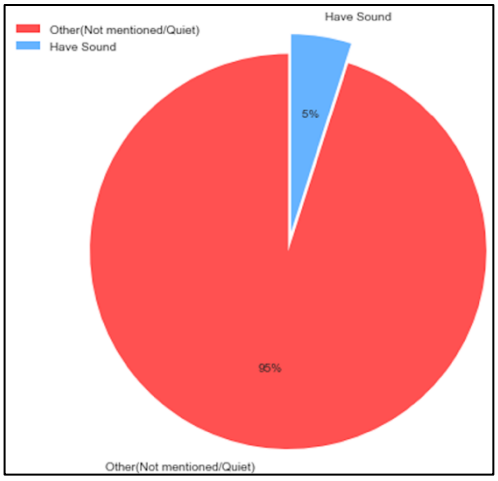


Figure 8
Generated by Code6

Finding the audio pattern of UFO requires 'Target Words Frequencies Counting' code which can search and count the number of descriptions containing particular words. For example, setting the target words as 'noise', 'noisy', 'hear' etc. (30~50 words related to hearing sound) can be applied to investigate the audio pattern of UFO. From figure 8, it can be concluded that approximately 5% of eyewitnesses claimed that they heard the sound from UFOs.

One advantage of the dataset is that all records have been sorted according to the time of UFO events. Hence, it is relatively easy to find the number of UFO events in a specific year or month by counting them in Microsoft Excel. It can be seen from Figure 9 that the number of UFO events has been on the rise since 1990 and reached its peak in 2014. After that, the number of sightings dropped sharply until 2018, and has been increasing thereafter. Figure 10 provides an intuitive overview of the number of UFO sightings in each month from 2009 to 2020. It can be concluded that people witnessed more UFOs than usual in July and August from 2012 to 2014 and March to April in 2020.

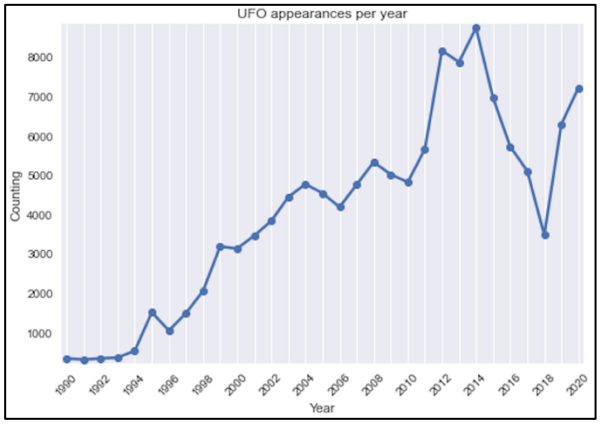


Figure 9
Generated by Code11

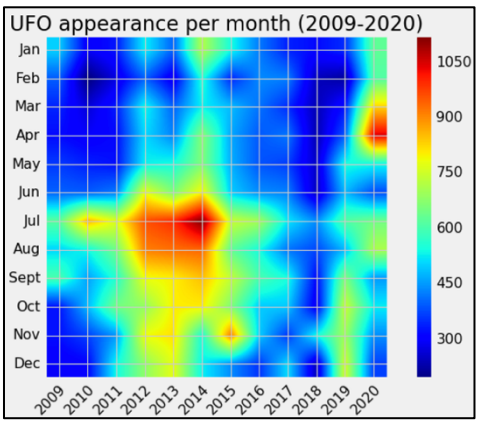


Figure 10
Generated by Code14

Finding the appearance time of UFO needs the 'Text Filter' from Microsoft Excel to avoid compatibility issues generated by Python. For example, to find the number of UFO sightings at 20:00 to 21:00, the content in the 'Text Filter' could be "contains '20:' ". It can be seen from Figure

11 that the number of UFO events increases sharply after 4 pm and it reaches a peak at 9 pm. After that, the number declines. This trend is consistent with previous speculation that the vast majority of people might witness UFOs at night.

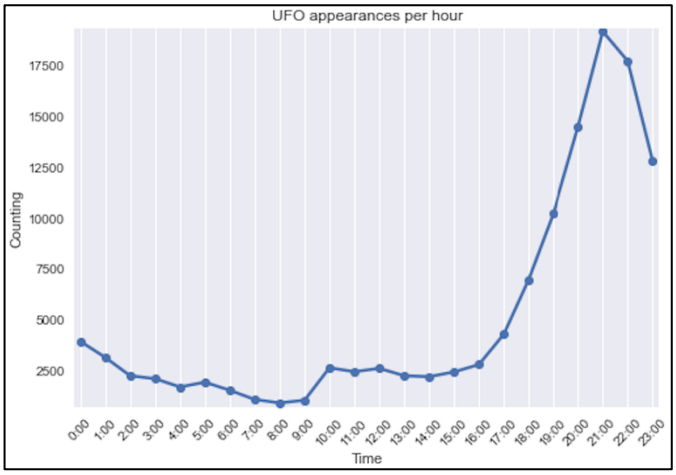


Figure 11

Generated by Code8

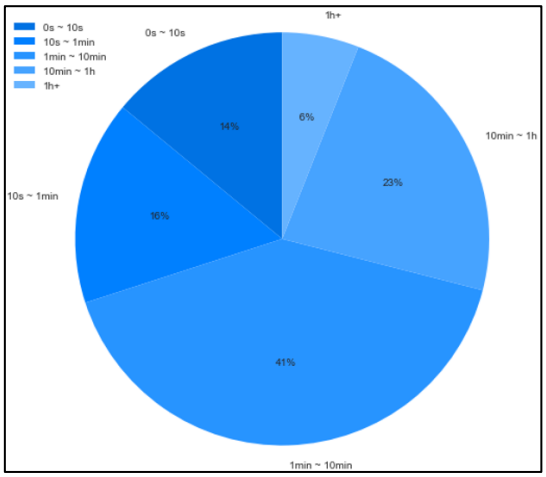


Figure 12

Generated by Code9

Considering the time constraint and actual situations, only data from 2020 were used to investigate the durations of UFO. Identifying this pattern requires Microsoft Excel combined with some manual work. Firstly, meaningless signs and English words were replaced by spaces. Then, empty records were deleted to make sure 'Text Filter' was able to work. Next, utilizing 'Text Filter' to filter various durations could generate outcomes. During the whole process, filtered data should be removed from the dataset after each round of searching to avoid repetition. It can be concluded from Figure 12 that there are 14% of UFOs will disappear within 10 seconds and 30% of UFOs will vanish within 1 minute. In addition, a vast majority of eyewitnesses (71%) can witness UFOs within 10 minutes.

Conclusion

In conclusion, this report provides people with an overview of the prevailing attributes of UFOs. Numbers of patterns have been identified and concluded from the descriptions of different eyewitnesses. However, the UFO phenomenon is still a mystery and needed to be studied continuously by human beings.

Summary of key findings

- Top 3 shape words describing UFOs are 'Circle', 'Triangle' and 'Fireball'.
- Top 3 color words describing UFOs are 'Bright', 'White' and 'Orange'.
- UFOs are most likely to be witnessed in 'California', 'Florida' and 'Washington'.
- People are more likely to witness UFOs from home, on the mountain, and on the street.
- Approximately 5% of eyewitnesses heard the sound from UFOs.
- The number of UFO sightings has risen since 1990 and reached its peak in 2014.
- People witnessed more UFOs in July, August from 2012 to 2014 and March, April in 2020.
- The number of UFO sightings increases sharply after 4 pm and reaches a peak at 9 pm.
- A vast majority of eyewitnesses (71%) can witness UFOs within 10 minutes.

Reference:

Carvalho, T 2020, *Heatmap Basics with Seaborn*, Towards Data Science, viewed 1 September 2021, <https://towardsdatascience.com/heatmap-basics-with-pythons-seaborn-fb92ea280a6c>.

Chauhan, A 2020*, Find frequency of each word in a string in Python*, GeeksforGeeks, viewed 10 August 2021,<https://www.geeksforgeeks.org/find- frequency-of-each-word-in-a-string-in-python/>.

Daniela, G et al. 2021, *Unidentified flying object*, Wikipedia, viewed 5 September 2021, <https://en.wikipedia.org/wiki/Unidentified_flying_object>.

Gawande, J 2021, *Bar Plot in Matplotlib*, GeeksforGeeks, viewed 21 August 2021, <https://www.geeksforgeek s.org/bar-plot-in-matplotlib/>.

Gawande, J 2020, *Plot a pie chart in Python using Matplotlib*, GeeksforGeeks, viewed 23 August 2021, < https://www.geeksforgeeks.org/plot-a-pie-chart-in-python-using-matplotlib/>.

Kaggle 2021, *National UFO Reporting Data*, csv, viewed 7 August 2021, <https://www.kaggle.com /connerbrew2/national-ufo-reporting-data>.

Knuth, K 2018, *Are we alone? The question is worthy of serious scientific study*, The Conversation, viewed 4 September 2021, <https:// theconversation.com/are-we-alone-the-question-is-worthy-of-serious-scientific-study-98843>.

Li, G 2021, *Make Waffle Charts in Python*, Python Repo, viewed 25 August 2021, <https://pythonrepo.com/repo/gyli-PyWaffle-python-data-validation>.

Line plot or line chart in python with legends 2021, Data Science Made Simple, viewed 26 August 2021, < https://www.datasciencemade simple.com/line-plot-line-chart-in-python-legends/>.

Stieb, M & Danner C 2021, *What's Inside the Pentagon's Long-Awaited UFO Report*, Intelligencer, viewed 5 September 2021, <https://nymag.com/intelligencer/article/pentagon-ufo-report-what-we-know.html>.

Vu, D 2019, *Generating Word Clouds in Python*, Data camp, viewed 29 August 2021, <https://www.datacamp.com/ community/tutorials/wordcloud-python>.

Wu, L 2020, *13 lines of Python code to draw a map of the US epidemic*, CSDN, viewed 27 August 2021, < https://blog.csdn.net/wulishinian /article/details/105539212>.

Appendix:

```
import pandas as pd
import numpy as np

df=pd.read_csv("./Desktop/Project1/UFO_Dateset.csv")

counts={}
for i in range(134421):
    txt=df.Summary.iloc[i]
    for ch in '#%,.?!"-()':
        txt=txt.replace(ch," ")
    txt=txt.lower()
    words=txt.split()

    thisRound={}
    for word in words:
        thisRound[word]=1

    for word in words:
        if thisRound[word]==2:
            break;
        else:
            thisRound[word]=2
            if word not in counts:
                counts[word]=1
            else:
                counts[word]+=1

# Ranking
items=list(counts.items())
items.sort(key=lambda x:x[1],reverse=True)
print("\n");
for i in range(5000):
    word,count=items[i]
    print("{0:<15}{1:>10}".format(word,count))
print("\n");
```

```
in                42446
the               37263
a                 33294
and               28976
lights            26242
of                25035
light             24715
sky               22892
object            19825
```

Code 1
All Words Frequencies Counting Code

```
import pandas as pd
import numpy as np

df=pd.read_csv("./Desktop/Project1/UFO_Dateset.csv")

count=0
target=['home','house','bedroom','kitchen',
        'houses','bedrooms','kitchens',]

for i in range(134421):
    txt=df.Summary.iloc[i]
    for ch in '#%,.?!"-()':
        txt=txt.replace(ch," ")
    txt=txt.lower()
    words=txt.split()

    allin=[]
    for word in words:
        allin.append(word)

    found=0

    for i in range(len(target)):
        if target[i] in allin:
            found=1

    if found==1:
        count=count+1

print("The number of records containing these words is :",
      count)
```

```
The number of records containing these words is : 4177
```

Code 2
Target Words Frequencies Counting Code

```
import numpy as np
import matplotlib.pyplot as plt
x = np.array(["Silver","Yellow","Dark","Black","Blue","Green","Red","Orange","White","Bright"])
y = np.array([1736,1820,2068,2676,4303,5149,10420,11737,11862,19535])

sortIndex = np.argsort(y) # Rank odering

x_sort = x[sortIndex] # Sorting
y_sort = y[sortIndex]

plt.xticks(np.arange(len(x_sort)), x_sort)
a = plt.bar(np.arange(len(x_sort)),y_sort,color=['#c5c9c7','#ffff14',
'#03012d','#1b2431', '#0343df', '#15b01a','#e50000','#f97306','#ffffe4','#ffffff'])

plt.title('Color Pattern of UFO')
plt.ylabel('Color Counting', fontsize=12)
plt.xlabel('Colors', fontsize=12)
plt.style.use('seaborn')
plt.grid(axis="y",ls='--')
plt.show()
```

Code 3
Bar plot for Color Pattern

```
import numpy as np
import matplotlib.pyplot as plt
y = np.array(["Flash","Changing","Cigar","Formation","Oval",
              "Disk","Sphere","Fireball","Triangle","Circle "])
x = np.array([2356,3262,3264,4479,5842,8002,8754,9516,12132,14163])


sortIndex = np.argsort(x) # Rank odering

x_sort = x[sortIndex] # Sorting
y_sort = y[sortIndex]

plt.yticks(np.arange(len(y_sort)), y_sort)
a = plt.barh(np.arange(len(y_sort)),x_sort,color=['#0080FF','#2894FF','#46A3FF',
        '#66B3FF', '#84C1FF', '#97CBFF','#ACD6FF','#C4E1FF','#D2E9FF','#ECF5FF'])

plt.plot(kind='barh')
plt.title('Shape Pattern of UFO')
plt.xlabel('Shape Counting', fontsize=12)
plt.ylabel('Shapes', fontsize=12)
plt.style.use('seaborn')
plt.grid(axis="x",ls='--')
plt.show()
```

Code 4
Bar plot for Shape Pattern

```
import numpy as np
import matplotlib.pyplot as plt
y = np.array(["Michigan","Illinois","Ohio","Pennsylvania","Arizona",
              "New York","Texas","Washington","Florida","California"])
x = np.array([3347,3979,4050,4500,4539,5286,5555,6617,7398,14863])

sortIndex = np.argsort(x) # Rank

x_sort = x[sortIndex] # Sorting
y_sort = y[sortIndex]

plt.yticks(np.arange(len(y_sort)), y_sort)
a = plt.barh(np.arange(len(y_sort)),x_sort,color=['#FFD2D2','#FFB5B5',
'#FF9797','#FF7575','#FF5151','#FF2D2D','#FF0000','#EA0000','#CE0000','#AE0000'])

plt.plot(kind='barh')
plt.title('State Positions of UFO')
plt.xlabel('State Positions Counting', fontsize=12)
plt.ylabel('States', fontsize=12)
plt.style.use('dark_background')
plt.grid(axis="x",ls='--')
plt.show()
```

Code 5
Bar plot for State Positions of UFO

```
import matplotlib.pyplot as plt
plt.figure(figsize=(7,7))
labels = [u'Other(Not mentioned/Quiet)', u'Have Sound']
sizes = [95.1,4.9]
colors = ['#FF5151', '#66B3FF']

explode = (0.1, 0)

patches, l_text, p_text = plt.pie(sizes, explode=explode, labels=labels, colors=colors,
                                  labeldistance=1.1, autopct='%2.0f%%', shadow=False,
                                  startangle=90, pctdistance=0.6)

for t in l_text:
    t.set_size = 30
for t in p_text:
    t.set_size = 20
plt.axis('equal')
plt.legend(loc='upper left', bbox_to_anchor=(-0.1, 1))
plt.grid()
plt.style.use('seaborn')
plt.show()
```

Code 6
Pie chart for Audio Pattern

```
from wordcloud import WordCloud

with open("./Desktop/word.txt" ,encoding="utf-8")as file:
    # Read text
    text=file.read()

# Some settings
wordcloud=WordCloud(font_path="./Library/Fonts/Times.ttc",
background_color="black",width=600,
height=400,max_words=100,relative_scaling=0.3).generate(text)
# Generate Word Clous Visualizations
image=wordcloud.to_image()
# Show this visualization
image.show()
```

Code 7
Word Cloud Visualization for Shape Pattern

```
import matplotlib.pyplot as plt
names = ['0:00','1:00','2:00','3:00','4:00','5:00','6:00','7:00','8:00','9:00',
         '10:00','11:00','12:00','13:00','14:00','15:00','16:00','17:00','18:00',
         '19:00','20:00','21:00','22:00','23:00']
x = range(len(names))
y = [3919,3131,2226,2083,1676,1911,1513,1061,892,1025,2618,2429,2590,
     2229,2181,2415,2778,4273,6962,10217,14460,19152,17704,12793]
plt.plot(x, y, linewidth=2.5,marker='o')
plt.legend()
plt.xticks(x, names, rotation=45)
plt.margins(0.01)

plt.subplots_adjust(bottom=0.15)
plt.xlabel(u"Time")
plt.ylabel("Counting")
plt.title("UFO appearances per hour")
plt.style.use('seaborn')
plt.grid(axis="y",ls='--')
plt.show()
```

Code 8
Line chart for Appearance time of UFO

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,8))
labels = [u'0s ~ 10s', u'10s ~ 1min', u'1min ~ 10min', u'10min ~ 1h', u'1h+']
sizes = [14,16,41,23,6]
colors = ['#0072E3','#0080FF','#2894FF','#46A3FF','#66B3FF']

explode = (0,0,0,0,0)

patches, l_text, p_text = plt.pie(sizes, explode=explode, labels=labels, colors=colors,
                                  labeldistance=1.1, autopct='%2.0f%%', shadow=False,
                                  startangle=90, pctdistance=0.6)

for t in l_text:
    t.set_size = 30
for t in p_text:
    t.set_size = 20
plt.axis('equal')
plt.legend(loc='upper left', bbox_to_anchor=(-0.1, 1))
plt.grid()
plt.style.use('seaborn')
plt.show()
```

Code 9
Pie chart for Durations of UFO

```
import numpy as np
import matplotlib.pyplot as plt
y = np.array(["Airport","Highway","Field","Sea","Beach","Backyard","Lake",
              "Street","Mountain","Home"])
x = np.array([738,1002,1014,1115,1154,1187,1592,1761,2448,4177])


sortIndex = np.argsort(x) # Rank odering

x_sort = x[sortIndex] # Sorting
y_sort = y[sortIndex]

plt.yticks(np.arange(len(y_sort)), y_sort)
a = plt.barh(np.arange(len(y_sort)),x_sort,color=['#FFD2D2','#FFB5B5',
'#FF9797','#FF7575','#FF5151','#FF2D2D','#FF0000','#EA0000','#CE0000','#AE0000'])

plt.plot(kind='barh')
plt.title('General Positions of UFO')
plt.xlabel('Positions Counting', fontsize=12)
plt.ylabel('Positions', fontsize=12)
plt.style.use('seaborn')
plt.grid(axis="x",ls='--')
plt.show()
```

Code 10
Bar plot for General Positions of UFO

```
import matplotlib.pyplot as plt
names = ['1990', ' ','1992',' ', '1994', ' '
         ,'1996',' ','1998',' ','2000',' ','2002',
         ' ','2004',' ','2006',' ','2008',' ','2010',
         ' ','2012',' ','2014',' ','2016',' ','2018',' ','2020']
x = range(len(names))
y = [332,303,331,351,527,1505,1038,1483,2045,3175,3124,
     3452,3826,4435,4764,4534,4179,4749,5323,5009,4814,
     5652,8161,7865,8739,6954,5720,5099,3469,6286,7213]

plt.plot(x, y, linewidth=2.5,marker='o')
plt.legend()
plt.xticks(x, names, rotation=45)
plt.margins(0.01)

plt.subplots_adjust(bottom=0.15)
plt.xlabel(u"Year")
plt.ylabel("Counting")
plt.title("UFO appearances per year")
plt.style.use('seaborn')
plt.grid(axis="y",ls='--')
plt.show()
```

Code 11
Line chart for Appearance year of UFO

```
import plotly.graph_objects as go
import pandas as pd

df=pd.read_excel("./Desktop/Project1/Research outcomereal.xlsx")

fig = go.Figure(data=go.Choropleth(
    locations=df['visualization state'],
    z = df['num4'].astype(float),
    locationmode = 'USA-states',
    colorscale = 'Reds',
    colorbar_title = "Frequency",
))

fig.update_layout(
    title_text = 'USA State Positions of UFO',
    geo_scope='usa',

)
```

Code 12
Map Visualization for State Positions of UFO

```
import matplotlib.pyplot as plt
from pywaffle import Waffle

plt.figure(
    FigureClass=Waffle,
    rows=10,
    columns=10,
    values=[26.92,16.35,16.17,14.36,7.10,5.93,
            3.69,2.85,2.51,2.39,1.06,0.68],
    colors=("#ffffff", "#ffffe4","#f97306","#e50000",
            "#15b01a","#0343df","#1b2431","#03012d",
            "#ffff14","#c5c9c7","#c44240","#fd3c06"),
)
plt.show()
```

Code 13
Waffle Visualization for Color Pattern

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
import seaborn as sns
import palettable
from sklearn import datasets

df=pd.read_csv("./Desktop/Project1/month.csv",index_col = ['month'])

df

months = ["Jan", "Feb", "Mar", "Apr","May", "Jun", "Jul","Aug","Sept","Oct","Nov","Dec"]

years = ["2009","2010","2011","2012","2013","2014","2015","2016","2017","2018","2019","2020"]

data = np.array([[501, 296, 333, 531, 401, 720, 548, 424, 340, 320, 351, 627],
                 [398, 195, 278, 400, 286, 560, 347, 438, 428, 246, 222, 612],
                 [344, 267, 338, 540, 401, 531, 483, 417, 350, 240, 329, 828],
                 [320, 297, 319, 509, 431, 673, 465, 392, 420, 236, 394, 1045],
                 [364, 339, 324, 532, 546, 663, 473, 388, 367, 256, 549, 541],
                 [400, 394, 423, 780, 661, 796, 499, 440, 428, 250, 480, 379],
                 [620, 862, 768, 959, 994, 1122, 746, 703, 522, 441, 611, 647],
                 [507, 541, 648, 913, 929, 938, 631, 546, 418, 377, 481, 713],
                 [603, 458, 562, 784, 802, 849, 741, 601, 548, 316, 665, 451],
                 [329, 482, 649, 686, 804, 805, 667, 496, 495, 286, 746, 528],
                 [327, 372, 451, 786, 815, 552, 913, 499, 361, 541, 699, 442],
                 [296, 311, 541, 681, 777, 531, 441, 358, 512, 246, 759, 370]])

fig, ax = plt.subplots()
im = ax.imshow(data)

# We want to show all ticks...
ax.set_xticks(np.arange(len(years)))
ax.set_yticks(np.arange(len(months)))
# ... and label them with the respective list entries
ax.set_xticklabels(years)
ax.set_yticklabels(months)

# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")

ax.set_title("UFO appearance per month (2009-2020)")
fig.tight_layout()
plt.colorbar(im)
plt.style.use('fivethirtyeight')
plt.show()
```

Code 14
Heat Map for Appearance Month of UFO

```
import random
import pandas as pd
import numpy as np

file_handle=open("./Desktop/word

index=0
words=[0 for x in range(0, 902)]
for i in range(5):
    words[index]="cross"
    index=index+1
for i in range(6):
    words[index]="cone"
    index=index+1
for i in range(13):
    words[index]="teardrop"
    index=index+1
for i in range(13):
    words[index]="egg"
    index=index+1
for i in range(17):
    words[index]="chevron"
    index=index+1
for i in range(21):
    words[index]="diamond"
    index=index+1
for i in range(23):
    words[index]="cylinder"
    index=index+1
for i in range(25):
    words[index]="rectangle"
    index=index+1
for i in range(26):
    words[index]="flash"
    index=index+1
for i in range(35):
    words[index]="changing"
    index=index+1
for i in range(35):
    words[index]="cigar"
    index=index+1
```

```
for i in range(26):
    words[index]="flash"
    index=index+1
for i in range(35):
    words[index]="changing"
    index=index+1
for i in range(35):
    words[index]="cigar"
    index=index+1
for i in range(49):
    words[index]="formation"
    index=index+1
for i in range(63):
    words[index]="oval"
    index=index+1
for i in range(87):
    words[index]="disk"
    index=index+1
for i in range(95):
    words[index]="sphere"
    index=index+1
for i in range(103):
    words[index]="fireball"
    index=index+1
for i in range(132):
    words[index]="triangle"
    index=index+1
for i in range(154):
    words[index]="circle"
    index=index+1

for i in range(10000):
    random.shuffle(words)

for i in words:
    file_handle.write(i)
    file_handle.write(" ")


file_handle.close()
```

Code 15
Generating Input for Word Cloud Visualization

```
le changing circle circle s
 teardrop triangle disk cyl
ircle fireball rectangle fo
 circle disk circle formati
k triangle cigar triangle f
e teardrop circle triangle
disk disk chevron circle ci
 formation triangle circle
 rectangle oval sphere circ
k diamond chevron formation
cle fireball circle cigar c
mation circle rectangle cir
iangle sphere fireball rect
ing oval circle cigar circl
cle triangle disk changing
iangle fireball fireball fi
hanging sphere triangle fla
riangle sphere fireball dia
riangle circle cylinder egg
disk sphere sphere changing
al disk triangle triangle d
e oval changing egg cone tr
ectangle circle sphere circ
 triangle formation sphere
cigar formation formation s
isk fireball triangle recta
rdrop disk triangle oval ov
ircle rectangle cigar fireb
 cross triangle diamond tri
ireball sphere circle trian
riangle oval egg fireball f
ngle chevron fireball fireb
ircle sphere changing circl
riangle cigar triangle fire
n changing diamond sphere s
cle oval fireball circle di
anging disk circle changing
 triangle cross circle tria
egg flash circle triangle c
diamond circle sphere disk
 cylinder oval disk diamond
rcle circle triangle disk c
hanging oval flash cylinder
isk triangle circle triangl
ctangle fireball disk cigar
ormation sphere circle disk
hanging oval sphere circle
e triangle oval cigar disk
ond triangle triangle circl
 formation disk sphere tria
riangle sphere fireball sph
all formation triangle flas
oval cigar fireball disk ch
riangle changing sphere for
ormation fireball triangle
 flash disk circle fireball
ormation fireball sphere fi
e flash triangle teardrop c
tion triangle triangle circ
ball circle cylinder disk t
isk triangle rectangle cros
ireball sphere circle cylin
l fireball fireball triangl
eball disk cylinder triangl
 triangle triangle circle f
 circle fireball circle dis
e triangle rectangle sphere
fireball disk oval triangle
 triangle oval teardrop cir
igar circle sphere circle f
```

Figure 13
Text File Generated by Code 15

Excel Worksheet

	A	B	C	D	E	F	G	H
1	Colors	Color Counting	Shape Pattern of UFO	num2	State Positions Pattern	num3	Audio Pattern of UFO	
2	silver	1736	cross	435	NJ	2621	Has sound 4.87%	6553
3	yellow	1820	cone	537	CO	2905	Not mentiond / no sound 95.13%	127826
4	dark	2068	teardrop	1163	OR	3264		
5	black	2676	egg	1171	NC	3285		
6	blue	4303	chevron	1537	MI	3347	Michigan	
7	green	5149	diamond	1948	IL	3979	Illinois	
8	red	10420	cylinder	2091	OH	4050	Ohio	
9	orange	11737	rectangle	2284	PA	4500	Pennsylvania	
10	white	11862	flash	2356	AZ	4539	Arizona	
11	bright	19535	changing	3262	NY	5286	New York	
12			cigar	3264	TX	5555	Texas	
13		0	formation	4479	WA	6617	Washington	
14			oval	5842	FL	7398	Florida	
15			disk	8002	CA	14863	California	
16			sphere	8754				
17			fireball	9516				
18			triangle	12132				
19			circle	14163				

	J	K	L	M	N	O	P	Q
1	visualization state	num4		year	num5		during	num6
2	AK	607		1990	332		0:00	3919
3	AL	1235		1991	303		1:00	3131
4	AR	1136		1992	331		2:00	2226
5	AZ	4539		1993	351		3:00	2083
6	CA	14863		1994	527		4:00	1676
7	CO	2905		1995	1505		5:00	1911
8	CT	1812		1996	1038		6:00	1513
9	DE	361		1997	1483		7:00	1061
10	FL	7398		1998	2045		8:00	892
11	GA	2469		1999	3175		9:00	1025
12	HI	630		2000	3124		10:00	2618
13	IA	1126		2001	3452		11:00	2429
14	ID	1226		2002	3826		12:00	2590
15	IL	3979		2003	4435		13:00	2229
16	IN	2451		2004	4764		14:00	2181
17	KS	1087		2005	4534		15:00	2415
18	KY	1538		2006	4179		16:00	2778
19	LA	1016		2007	4749		17:00	4273
20	MA	2400		2008	5323		18:00	6962
21	MD	1655		2009	5009		19:00	10217
22	ME	1087		2010	4814		20:00	14460
23	MI	3347		2011	5652		21:00	19152
24	MN	1935		2012	8161		22:00	17704
25	MO	2568		2013	7865		23:00	12793
26	MS	729		2014	8739			122238
27	MT	937		2015	6954			
28	NC	3285		2016	5720			
29	ND	259		2017	5099			
30	NE	647		2018	3469			
31	NH	1073		2019	6286			
32	NJ	2621		2020	7213			
33	NM	1511						
34	UT	1389						
35	NV	1567						
36	NY	5286						
37	OH	4050						
38	OK	1340						
39	OR	3264						
40	PA	4500						
41	RI	556						
42	SC	2060						
43	SD	351						
44	TN	2068						
45	TX	5555						
46	VA	2444						
47	VT	549						
48	WA	6617						
49	WI	2261						
50	WV	828						
51	WY	379						

Note: Yellow entries represent coastal states in the USA

	AB	AC	AD	AE	AF	AG
1	cross	435	0.004719745	4.719744808	5	cross
2	cone	537	0.005826444	5.826443591	6	cone
3	teardrop	1163	0.012618536	12.61853612	13	teardrop
4	egg	1171	0.012705336	12.70533602	13	egg
5	chevron	1537	0.016676432	16.67643166	17	chevron
6	diamond	1948	0.021135777	21.13577675	21	diamond
7	cylinder	2091	0.022687325	22.68732504	23	cylinder
8	rectangle	2284	0.024781373	24.78137274	25	rectangle
9	flash	2356	0.025562572	25.56257188	26	flash
10	changing	3262	0.035392661	35.39266107	35	changing
11	cigar	3264	0.035414361	35.41436104	35	cigar
12	formation	4479	0.048597097	48.59709654	49	formation
13	oval	5842	0.06338563	63.38563028	63	oval
14	disk	8002	0.086821604	86.8216045	87	disk
15	sphere	8754	0.094980796	94.98079552	95	sphere
16	other	9230	0.10014539	100.1453898	100	other
17	fireball	9516	0.103248486	103.2484864	103	fireball
18	triangle	12132	0.131632055	131.6320552	132	triangle
19	circle	14163	0.153668381	153.668381	154	circle

Note: Numbers in AF column represent the relative numbers for shape words.

AQ	AR	AS	AT	AU
red/orange	492	0.006780128	0.678012816	#fd3c06
reddish	767	0.010569834	1.056983394	#c44240
silver	1736	0.023923379	2.392337904	#c5c9c7
yellow	1820	0.025080962	2.50809619	#ffff14
dark	2068	0.028498587	2.849858747	#03012d
black	2676	0.036877282	3.687728244	#1b2431
blue	4303	0.05929856	5.929855991	#0343df
green	5149	0.070957073	7.095707297	#15b01a
red	10420	0.143595397	14.35953972	#e50000
orange	11737	0.161744643	16.17446427	#f97306
white	11862	0.163467236	16.34672363	#c6fcff
bright	19535	0.269206918	26.92069179	#ffffff

Note: Numbers in AT column represent the ratio of colors

AX	AY	AZ
10s]	789	0.136198861
10s-1min]	941	0.162437424
1min-10mi	2373	0.409632315
10min-1h]	1357	0.234248231
1h+	333	0.057483169
sum	5793	

Note: Durations of UFO (2020)

BD	BE	BF
place	num	
home / house / bedroom / kitchen	4177	Home
valley / foothills / mountains / hill	2448	Mountain
crossing / road /street / parkway	1761	Street
lake	1592	Lake
backyard / yard / garden	1187	Backyards
beach	1154	Beach
ocean / sea /	1115	Sea
field / farm	1014	Field
highway / freeway	1002	Highway
airport	738	Airport
coast / gulf / shore	710	
park	704	
island	548	
woods / forest / grove	518	
school / college university	475	
porch	390	
station	286	
deck	268	
desert	242	
port	156	
creek / stream	261	
canyon	118	
patio	104	
pool	96	
camp	97	

BA	BB
2008 1	477
2	373
3	343
4	445
5	349
6	494
7	581
8	514
9	391
10	540
11	463
12	352

2009 1	501
2	398
3	344
4	320
5	364
6	400
7	620
8	507
9	603
10	329
11	327
12	296

2010 1	296
2	195
3	267
4	297
5	339
6	394
7	862
8	541
9	458
10	482
11	372
12	311

2011 1	333
2	278
3	338
4	319
5	324
6	423
7	768
8	648
9	562
10	649
11	451
12	541

2012 1	531
2	400
3	540
4	509
5	532
6	780
7	959
8	913
9	784
10	686
11	786
12	681

2013 1	401
2	286
3	401
4	431
5	546
6	661
7	994
8	929
9	802
10	804
11	815
12	777

2014 1	720
2	560
3	531
4	673
5	663
6	796
7	1122
8	938
9	849
10	805
11	552
12	531

2015 1	548
2	347
3	483
4	465
5	473
6	499
7	746
8	631
9	741
10	667
11	913
12	441

2016 1	424
2	438
3	417
4	392
5	388
6	440
7	703
8	546
9	601
10	496
11	499
12	358

2017 1	340
2	428
3	350
4	420
5	367
6	428
7	522
8	418
9	548
10	495
11	361
12	512

2018 1	320
2	246
3	240
4	236
5	256
6	250
7	441
8	377
9	316
10	286
11	541
12	246

2019 1	351
2	222
3	329
4	394
5	549
6	480
7	611
8	481
9	665
10	746
11	699
12	759

2020 1	627
2	612
3	828
4	1045
5	541
6	379
7	647
8	713
9	451
10	528
11	442
12	370

Data for appearance month of UFO