UFO SIGHTINGS

Ali Krakowsky

INTRODUCTION

Do you remember our Pandas class from July?

There was a topic I thought would be interesting to apply some of things we've learned since then to expand the data.

…

UFO Sightings!

SUMMARY

NUFORC is the National UFO Reporting Center where the reports of UFO sightings are stored. The goal of this project is to pull all of the reports and create visuals using Plotly.

SOURCES

1. National UFO Reporting Center (nuforc.org) list of ufo sightings from 9/2021 and previous years
2. https://github.com/kelvins/US-Cities-Database/blob/main/csv/us_cities.csv provides a list of US cities and states with the latitude and longitude.
3. https://www.youtube.com/watch?v=B97qWOUvlnU Code with Prince provided a video tutorial to add interactive charts to a flask app

PROCESS

- First step- gather the data
- To pull the reports from the site Jupyter Notebook was used

UFO Sightings

```
]:  # Dependencies
    from bs4 import BeautifulSoup
    import requests
    import pymongo
    from splinter import Browser
    from webdriver_manager.chrome import ChromeDriverManager
    import pandas as pd
    import time
```

```
]:  # Initialize PyMongo to work with MongoDBs
    conn = 'mongodb://localhost:27017'
    client = pymongo.MongoClient(conn)
```

```
]:  # Define database and collection
    db = client.ufo_db
    collection = db.ndxevent
```

```
]:  # URL of page to be scraped
    url = 'http://www.nuforc.org/webreports/ndxevent.html'
```

```
]:  # Retrieve page with the requests module
    response = requests.get(url)
    response.text
```

```
t[10]: '<HTML>\r\n<HEAD>\r\n<META HTTP-EQUIV="Content-Type" CONTENT="text
```

PROCESS

- From the site, the page Event Date was used
- Executable path created
- Searched for table data
- Looped through each link to create the data frame
- Result = Data pulled from almost 1,000 links

```
executable_path = {'executable_path': ChromeDriverManager().install()}
browser = Browser('chrome', **executable_path, headless=False)

url = 'http://www.nuforc.org/webreports/ndxevent.html'
browser.visit(url)
```

```
====== WebDriver manager ======
Current google-chrome version is 94.0.4606
Get LATEST driver version for 94.0.4606
Driver [C:\Users\alig_\.wdm\drivers\chromedriver\win32\94.0.4606.61\chromedriver.
```

```
data = browser.find_by_css("td a")
```

```
ufo_links = [x["href"] for x in data]
```

```
browser.quit()
```

```
df_list = []
for index,i in enumerate(ufo_links):
    df = pd.read_html(i)[0]
    df_list.append(df)
    print(index)
    time.sleep(1)
```

```
0
1
```

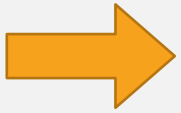
PROCESS

NUFORC Site

National UFO Reporting Center
Report Index by Month
Click on links for details

NUFORC Home

Reports	Count
10/2021	95
09/2021	223
08/2021	238
07/2021	177
06/2021	200
05/2021	458



National UFO Reporting Center
Monthly Report Index For 09/2021
Click on links for details

NUFORC Home

Date / Time	City	State	Shape	Duration	
9/30/21 22:50	Ocala	FL		45 seconds	Object trave
9/30/21 22:49	Atlanta	GA	Fireball	2 minutes	Maybe a me
9/30/21 21:45	Lakeland	GA	Other	60 seconds	Straight light
9/30/21 21:25	Grand Haven	MI	Light	01:00	Single, Brigh
9/30/21 20:59	Lewis Center	OH	Triangle	5 minutes	Traveling eas
9/30/21 20:40	Fenton	MI	Oval	90 seconds	Bright white
9/30/21 20:30	Los Angeles	CA	Circle	10 seconds	Two bright s
9/30/21 19:02	Franklin	KY			MADAR Nod
9/30/21 16:18	Whittier	CA	Changing	3 minutes	Today Septe

PROCESS

- Data frame created
- US_cities.csv added and merged to add lat and long
- Data frame saved as csv

```
ufo_locations = ufo_sightings.merge(cities, how="inner", left_on=["State", "Ci
ufo_locations
```

7]:

	Date / Time	City	State	Shape	Duration	Summary	Posted	Location
0	9/17/21 22:10	Laguna Hills	CA	Light	15 minutes	At 10:10 pm I walked outside, scattered clouds...	NaN	Laguna Hills, CA
1	11/11/20 16:13	Laguna Hills	CA	Circle	13 minutes	It lasted for 13 minutes, moved and then disap...	12/23/20	Laguna Hills, CA
2	11/11/20 16:13	Laguna Hills	CA	Circle	13 minutes	Red lights were going inside two red crafts	12/23/20	Laguna Hills, CA
3	8/18/19 21:45	Laguna Hills	CA	Circle	30	Orange light seen. In the blink of an eye it d...	8/23/19	Laguna Hills, CA
4	7/7/16 21:47	Laguna Hills	CA	Circle	2:59 seconds	Starlike object observed.	7/15/16	Laguna Hills, CA
...	...	...	...	...	...	...	...	...

DATA CLEANUP

- Prior to merging the csvs
  - The city and state were combined to a new column(Locations)
  - All sightings that were missing the location were dropped
  - Canadian sightings were dropped due to variation in data entry
- After cleaning- over 100,000 rows were left

```
ufo_sightings['Location'] = ufo_sightings['City'] + ", " + ufo_sightings['State']
ufo_sightings
```

Date / Time	City	State	Shape	Duration

```
ufo_sightings = ufo_sightings.dropna(how="all", subset=["Location"])
ufo_sightings
```

Date / Time	City	State	Shape	Duration
0 9/17/21 22:10	Laguna Hills	CA	Light	15 minutes	At 10:10 pm I well

VISUAL STUDIO CODE

- App.py was made to create a database in MongoDB
  - Able to convert the data to json file
  - Easier to review the labels when creating the flask app

```python
from flask import Flask, render_template, jsonify
from flask_pymongo import PyMongo

app = Flask(__name__)

mongo = PyMongo(app, uri="mongodb://localhost:27017/ufo_db")

@app.route('/')
def db_ping():

    return render_template('index.html', data = 'UFO Sightings')


@app.route('/data')
def db_data():

    db_data = mongo.db.ufo_sightings.find({}, {'_id': False})
    print('this route was pinged')
    parsed = [x for x in db_data]
    # print('parsed: ', parsed)
    return jsonify(parsed)
        You, 4 days ago • update files
if __name__ == '__main__':
    app.run(debug=True)
```

VISUAL STUDIO CODE

- Main.css: stylesheet to format the landing page
  - Font files saved as Web Open Font Format(WOFF) used to change the default font style

```
@font-face {
  font-family: "lobsterregular";
  src: url("./fonts/lobster-regular-webfont.woff2"
    url("./fonts/lobster-regular-webfont.woff") fo
  font-weight: bold;
  font-style: normal;
}

@font-face {
  font-family: "work_sansregular";
  src: url("./fonts/worksans-variablefont_wght-web
    url("./fonts/worksans-variablefont_wght-webfon
  font-weight: normal;
  font-style: normal;
}

* {       You, a day ago • file updates
  padding: 0px;
  margin: 0;
  font-family: work_sansregular;
}

.navbar-text {
  color: rgb(250, 250, 250);
}

.navbar-brand {
  color: rgb(250, 250, 250);
  font-family: lobsterregular;
  font-size: 2.5rem;
```

VISUAL STUDIO CODE

- Creating the flask app run.py
  - url_for to grab the font source
  - Plotly Express interactive visual tool

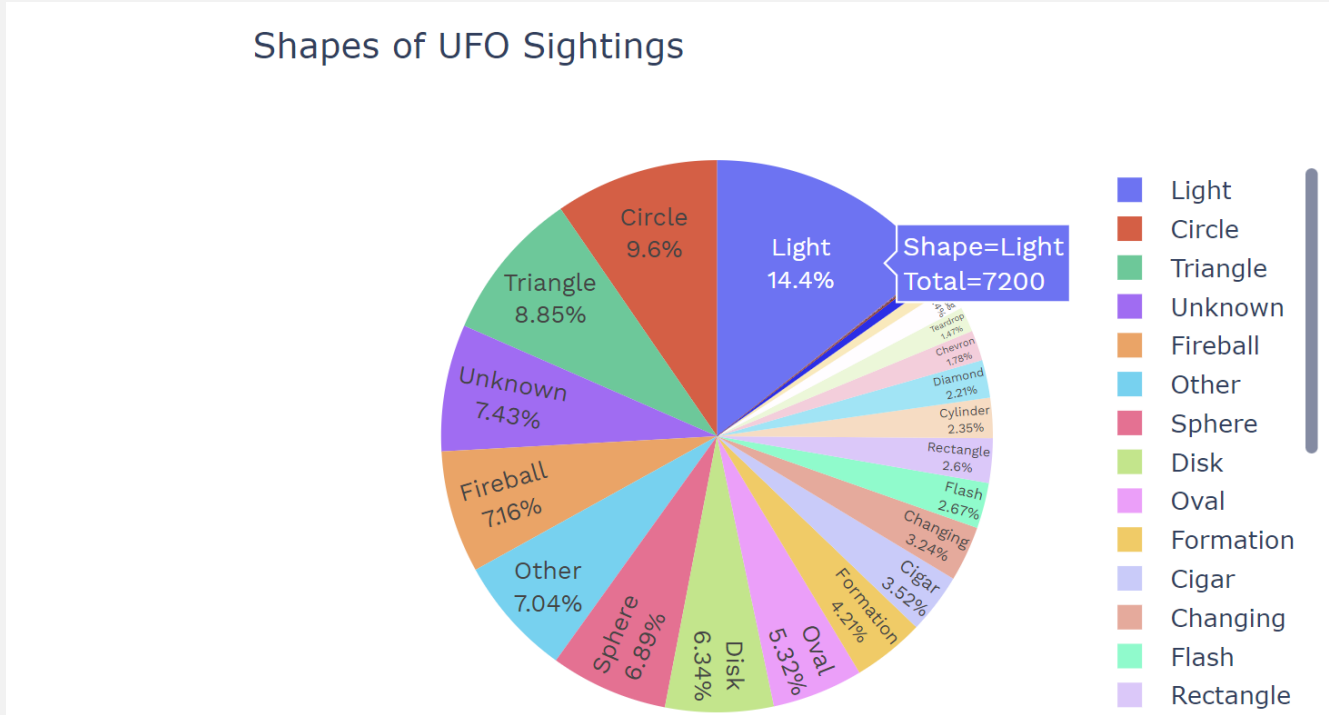
```
from flask import Flask, render_template, jsonify, url_for
import pandas as pd
import json
import plotly
import plotly.express as px

app = Flask(__name__)

@app.route("/")
def index():

    # Insert Visual 1
```

VISUAL #1

- Grouped the sightings by shapes
- Removed any sightings less than 5
- Created a pie chart with the name and percent inside the wedge
- Json.dumps- creates a trace to pass the data through as html

```
#Create the pie chart
fig = px.pie(df, values='ID', names='Shape',
             title='Shapes of UFO Sightings',
             hover_data=['ID'], labels={'ID':'Total'})
fig.update_traces(textposition='inside', textinfo='percent+label')
fig1JSON = json.dumps(fig, cls =plotly.utils.PlotlyJSONEncoder)
```

VISUAL #1



This pie chart is showing the reported shapes of the UFO sightings with the count and percentage. The most popular shape is Light. Sighting reports were created by viewers and submitted as free-text. This causes a variety in the data provided.

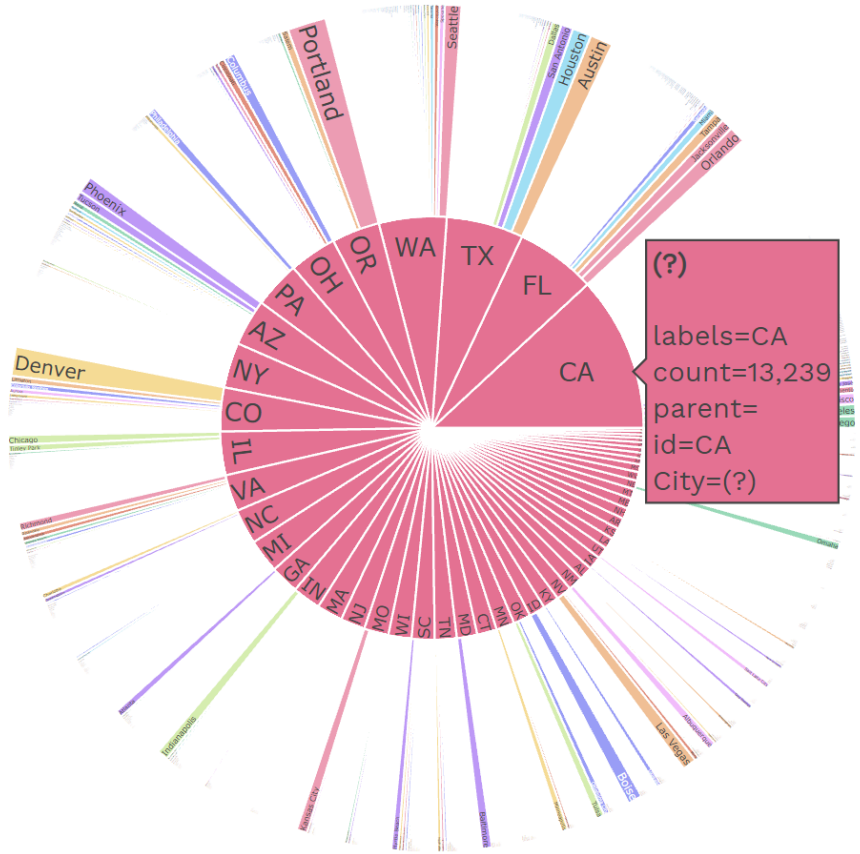
VISUAL #2

- Used the original dataframe
- Plotly express sunburst with State in the center and City on the outside
- Converted to html

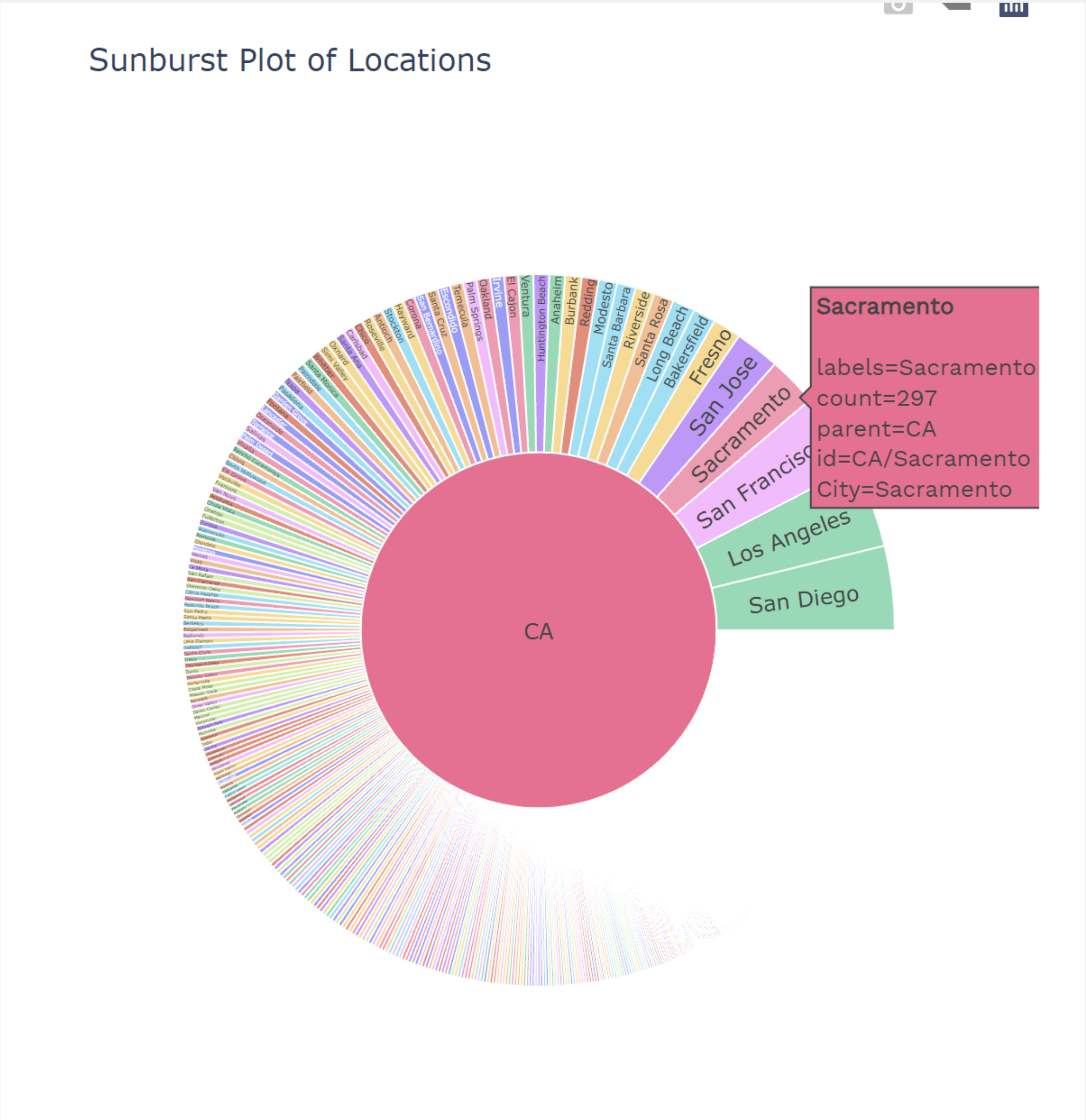
```
sun = px.sunburst(sun_df, title='Sunburst Plot of Locations', path=["State","City"],
hover_name="City", color="City", height=700)
#sun.show()
fig2JSON = json.dumps(sun, cls =plotly.utils.PlotlyJSONEncoder)
```

VISUAL #2

Sunburst Plot of Locations



VISUAL #2

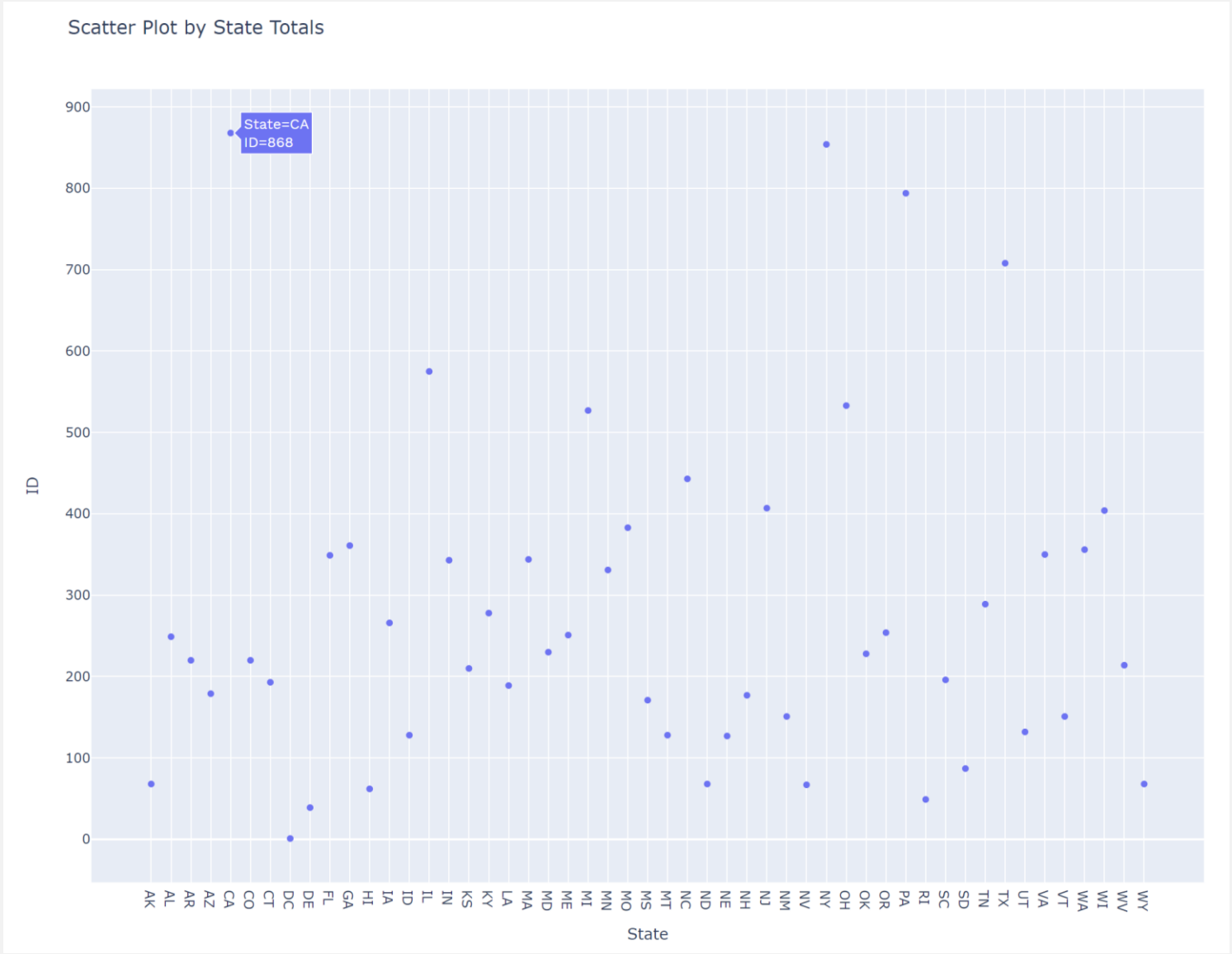

Sunburst Plot of Locations
CA
San Diego
Los Angeles
San Francisco
Sacramento
San Jose
Fresno
Bakersfield
Long Beach
Santa Rosa
Riverside
Santa Barbara
Modesto
Redding
Burbank
Anaheim
Huntington Beach
Ventura
El Cajon
Irvine
Oakland
Palm Springs
Sacramento
labels=Sacramento
count=297
parent=CA
id=CA/Sacramento
City=Sacramento

VISUAL #3

- Scatter plot of the totals in each state
- Groupby State and total found by each unique ID
- Plotly express creates a pop-up for each point

```
scatter = px.scatter(state_df, x="State", y="ID", title="Scatter Plot by State Totals", height=900)
#scatter.show()
fig3JSON = json.dumps(scatter, cls =plotly.utils.PlotlyJSONEncoder)

return render_template("index.html", fig1JSON = fig1JSON, fig2JSON = fig2JSON, fig3JSON = fig3JSON)
```

VISUAL #3

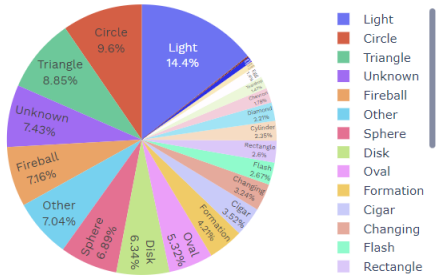

Scatter Plot by State Totals
State=CA
ID=868
ID
900
800
700
600
500
400
300
200
100
0
AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY
State

INDEX

- Used bootstrap to create a container and place each visual and add a small description for each
- Added a script to keep the visuals interactive
  - Each visual is added as a variable
  - Fig#JSON pulled from run.py

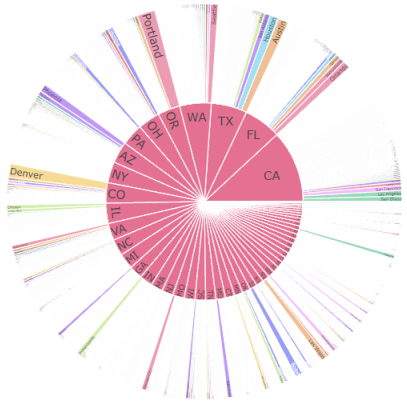
```
<script src="https://cdn.plot.ly/plotly
<script type="text/javascript">
    var fig1 = {{ fig1JSON | safe}};
    Plotly.plot("chart1", fig1,{});

    var fig2 = {{ fig2JSON | safe}};
    Plotly.plot("chart2", fig2,{});

    var fig3 = {{ fig3JSON | safe}};
    Plotly.plot("chart3", fig3,{});

</script>
</body>
```

Shapes of UFO Sightings



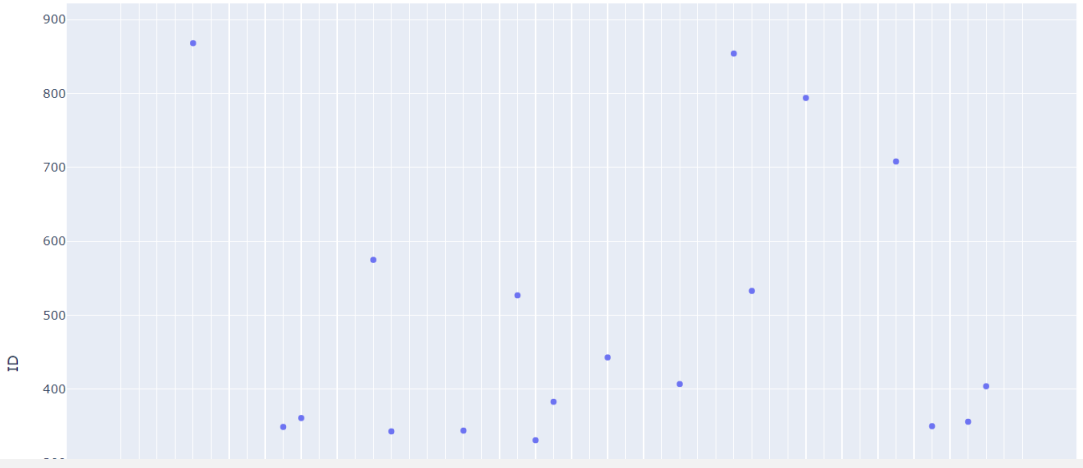
This pie chart is showing the reported shapes of the UFO sightings with the count and percentage. The most popular shape is Light. Sighting reports were created by viewers and submitted as free-text. This causes a variety in the data provided.

Sunburst Plot of Locations



The sunburst chart shows all of the states in the center and the cities on the outer edge. The states and cities are ordered from most to least sightings reported counter-clockwise. The state can be selected to show an expanded view of the state and the cities.

Scatter Plot by State Totals



SETBACKS

- Scraping the data-find the right value to scrape each entry
- Scraping took ~40 minutes to get all of the entries
- Originally tried to use geopy to convert the locations to geo locations, but it timed out after 400 entries
- Original plan was to use leaflet and create a heatmap to visualize common location-set-up worked the first time, but when trying to run the app the following day the map was off
- When the Plotly Express visuals were first added they were static

DISCUSSION

The data could have used some cleaning to help with standardizing the values. The site receives the information from the UFO viewer and it is free text. Reporter is not required to fill in all fields. Lost 25,000 entries due to missing locations and Canadian entries.

GOING FORWARD

- Better to start off with a small sample set then once the systems were working run it with a larger data set
- Would like to look at the heatmap again

QUESTIONS?