INSTITUTO FEDERAL
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

Gustavo William
Hugo César Alves da Silva
João Paulo Dantas
Polyana Cristina Sousa

RELATÓRIO DE PRÁTICA INTEGRADA DE CIÊNCIA DE DADOS E APRENDIZADO DE MÁQUINA

Brasília - DF
28/07/2021

Sumário

1. Objetivos

Os objetivos desta pesquisa consistem em aplicar métodos e saberes advindos das áreas de ciências de dados e aprendizagem de máquina com o intuito de coletar um conjunto de dados do meio web, mais especificamente da plataforma Nuforc, de forma tabular, sendo que, dentro deste conjunto foram considerados os relatos registrados na plataforma cujo o período de ocorrência está na faixa temporal de setembro de 1997 e agosto de 2017, aproximadamente 20 anos.

Nesta etapa da sprint, o objetivo foi, a partir dos dados coletados construir artefatos que possibilitasse uma melhor visualização dos dados, bem como sua interpretação. Desse modo, o propósito é fazer com que os dados sejam analisados através de mapas e gráficos temáticos.

2. Descrição do problema

Para se analisar os dados coletados de uma forma visual e mais prática, deve-se plotar os mesmos em forma de gráficos e mapas. O problema então consiste em construir meios que facilitem a análise dos dados.

Para isso esta etapa do projeto consiste em:

1. Desenvolver um gráfico de barras agrupadas;
2. Desenvolver um gráfico de barras empilhadas;
3. A manipulação dos dados para tirar as informações desnecessárias e limpeza de dados informações;
4. Enriquecer os dados construindo mapas temáticos;
5. Criar um mapa do país inteiro (EUA) e plotar no mapa as ocorrências para todas as cidades;
6. Criar um mapa apenas do estado da Califórnia, para analisar se essas visualizações se distribuem de forma homogênea dentro do estado;
7. Identificar onde na Califórnia está localizada a maior quantidade de visualizações de objetos voadores não identificados;

3. Desenvolvimento

Para esta etapa do projeto os dados foram tratados para que seu uso fosse possibilitado. Para confecção dos gráficos foram realizadas consultas agrupadas nos dados para que fosse possível sua divisão e agrupamento, para as consultas usou-se informações sobre o estado onde houve mais ocorrência, bem como pelo formato do objeto relatado.

Para a confecção dos mapas, os dados precisaram passar por um tratamento e melhor adequação para remover registros com valores “vazios”, “nulos” e remover variáveis irrelevantes para a análise. Já para realmente construir os mapas com os dados tratados, foi utilizada a biblioteca *zipcode* para acessar as latitudes e longitudes necessárias para plotar as localidades nos mapas e o desenvolvimento dos mapas foi possível utilizando a biblioteca *gmaps.*

As tecnologias utilizadas no projeto foram a linguagem de programação PYTHON e o ambiente de desenvolvimento do Google Collaboratory, sendo as bibliotecas e API, as seguintes:

- Zipcode: permite acessar as latitudes e longitudes de dados;
- gmaps: API da Google que permite a criação de mapas em python;
- Folium: permite mostrar a magnitude de dados através das cores em duas dimensões.

Após a criação dos gráficos e mapas, foi feita a análise dos dados a fim de responder aos problemas definidos nesta etapa do projeto.

3.1 Respostas aos problemas da pesquisa

Analisar se as visualizações se distribuem de forma homogênea dentro do estado da Califórnia.

Pode-se observar que a distribuição dos relatos não ocorre de forma homogênea no estado da Califórnia. Há duas maiores concentrações de visualizações, ambas próximas ao litoral, mas uma maior mais ao norte e outra mais ao sul. Percebe que as maiores concentrações de avistamentos são feitos justamente próximos às maiores cidades do estado: São Francisco e Los Angeles. Conforme pode ser verificado no mapa de pontos de ocorrência na Figura 1 abaixo, as duas maiores concentrações de avistamentos são próximas à região das duas cidades. O que pode

explicar essa situação é o fato de que por serem cidades com grande quantidade populacional isso influencia na quantidade de pessoas que podem presenciar um fenômeno e assim relatá-los.

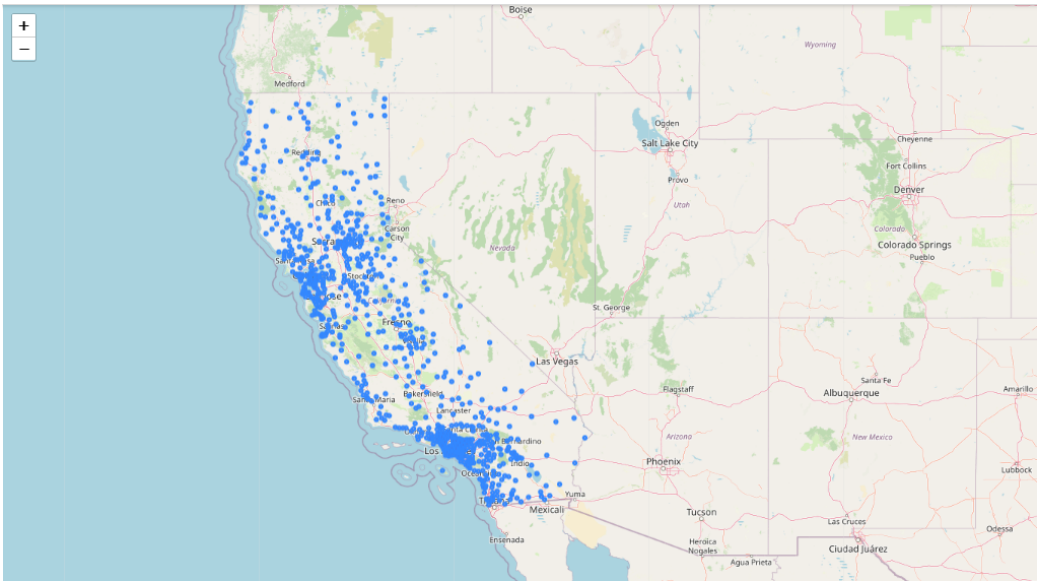


Figura 1: Mapa de pontos de ocorrência no estado da Califórnia.

Pela análise de dados e pela imagem do mapa de pontinhos da Figura 1 acima pode-se perceber que dentro do estado da Califórnia as maiores concentrações de registros são na região sudoeste do estado, mais próximo à cidade de Los Angeles. A justificativa para tal concentração se deve à cultura de avistamentos característico do estado e pelo fato de Los Angeles ser uma das maiores cidades e assim apresentando maiores possibilidades de avistamentos e relatos.

3.2 Código implementado

Sprint 2.1 - Criação de Gráficos e Mapas

Leitura do CSV e renomear colunas

```
f_data = pd.read_csv('https://raw.githubusercontent.com/infocbra/pratica-integrada-cd-e-am-2021-1-g1-ghjp/a496346889e5ce4d058ef7675ff853e4d344fe6d/1_sprint/ovinis_data.csv?token=AO5C2EBETFG6VUNLXT2SAT3BD5JGU')
df_data.columns = ['indexes','date/time', 'city', 'state', 'shape', 'duration', 'summary', 'posted']
df_data.drop(labels='indexes', axis=1, inplace=True)
```

Agrupar os resultados

```
shapes = df_data[['shape']].groupby(['shape']).size().sort_values(ascending=False)[:4].reset_index(level=0)
```

```
states = df_data[['state']].groupby(['state']).size().sort_values(ascending=False)[:4].reset_index(level=0)
```

Filtrar apenas os valores dos estados e formatos a serem utilizados

```
data = df_data.loc[df_data['shape'].isin(shapes['shape']) & df_data['state'].isin(states['state'])]
data = data[['shape', 'state']]
data = data.value_counts().reset_index()

data.columns = ['shape', 'state', 'total']
```

Estruturar o dataframe para facilitar o plot

```
data = data.set_index('state')
data = data.pivot(columns='shape')
data.columns = data.columns.droplevel()

data = data[['Light', 'Circle', 'Fireball', 'Triangle']]

Gráfico de barras agrupadas
data.plot(kind='bar')
```

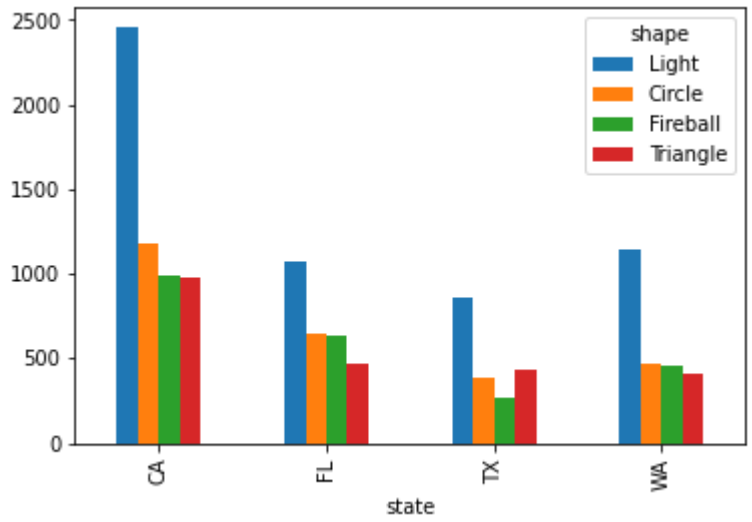
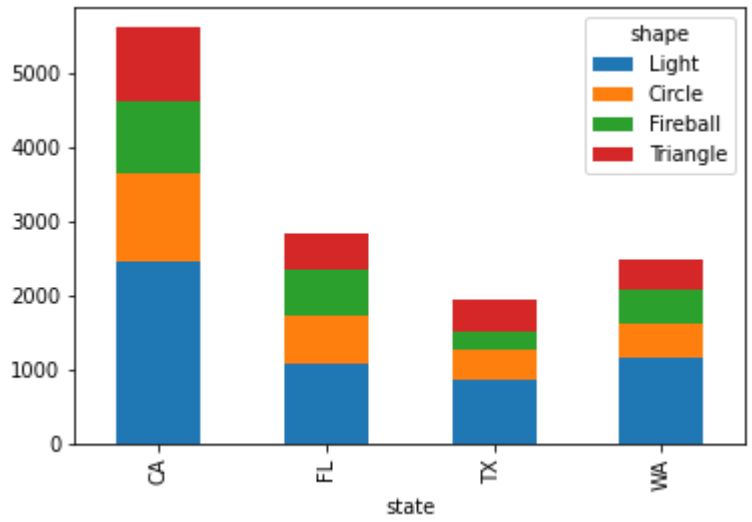


Gráfico de barras empilhadas

```
data.plot(stacked=True, kind='bar')
```



Assim, esses foram os passos e códigos para a criação dos gráficos de análise solicitados. Para a criação dos mapas de calor os passos em código foram os seguintes.

PARA CRIAÇÃO DOS MAPAS DE CALOR

Para importação de bibliotecas e API necessários para a criação dos mapas

```
import pandas as pd
import folium
from folium.plugins import HeatMap
from ipywidgets.embed import embed_minimal_html

!pip install zipcodes
import zipcodes
!pip install gmaps
import gmaps
```

Ler o arquivo CSV e renomear as colunas

```
df_data =
pd.read_csv('https://raw.githubusercontent.com/infocbra/pratica-integrada-cd-e-am-2021-1-g1-ghjp/a496346889e5ce4d058ef7675ff853e4d344fe6d/1_sprint/ovinis_data.csv?token=AO5C2EBETFG6VUNLXT2SAT3BD5JGU')
df_data.columns = ['indexes','date/time', 'city', 'state', 'shape', 'duration', 'summary', 'posted']
df_data.drop(labels='indexes', axis=1, inplace=True)
```

Utilizando funções da biblioteca zipcode

```
zipcodes_json = zipcodes.list_all()
df_cities = pd.DataFrame(zipcodes_json)[['city', 'state', 'lat', 'long']]
df_cities.drop_duplicates(['state', 'city'], inplace=True)
```

Visualizando a quantidade de ocorrência por localidade

```
data = df_data.groupby(['state', 'city']).size().reset_index()
data.columns = ['state', 'city', 'views']
```

Juntando os dataframes

```
coordinates = pd.merge(data, df_cities, how='inner', on=['state', 'city'])
coordinates['lat'] = pd.to_numeric(coordinates['lat'])
coordinates['long'] = pd.to_numeric(coordinates['long'])
```

Transformando os dados das colunas em listas de listas

```
plot_data_eua = [[row['lat'],row['long'], row['views']] for index, row in
coordinates.iterrows()]
```

Filtrando os dados do estado da Califórnia

```
cali_df = coordinates.loc[coordinates['state'] == 'CA']
```

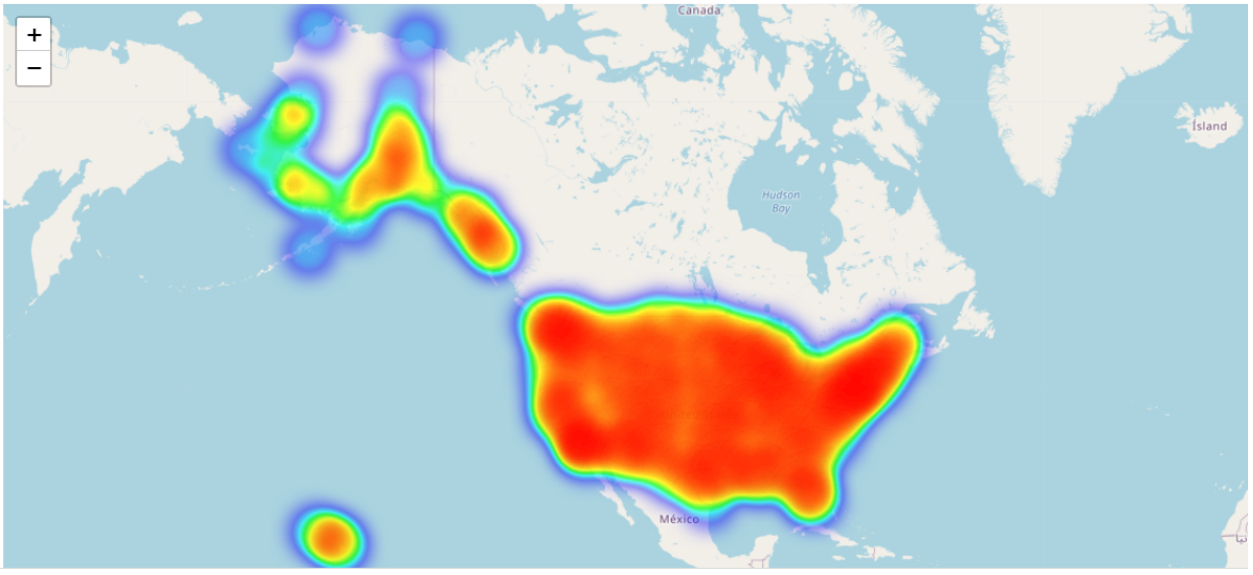
Transformar os dados das colunas em listas de listas

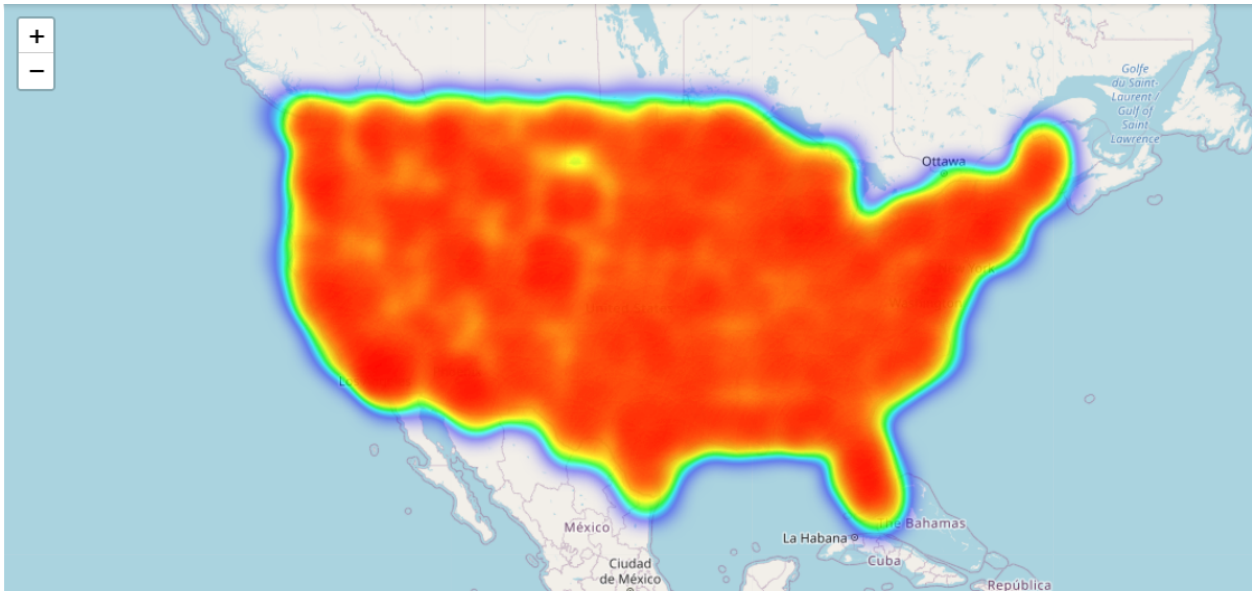
```
plot_data_cali = [[row['lat'],row['long'], row['views']] for index, row in
cali_df.iterrows()]
```

Mapas com Folium

Mapa de Calor dos EUA

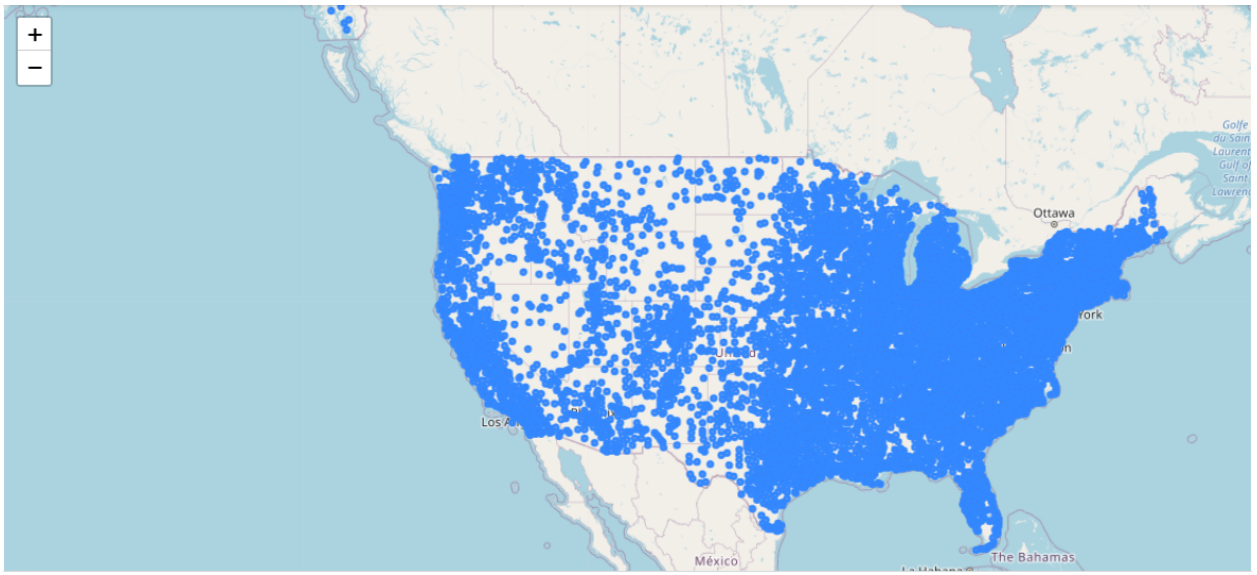
```
heatMapEUA = folium.Map(location=[39, -92], zoom_start = 3,
prefer_canvas=True)

HeatMap(plot_data_eua).add_to(heatMapEUA)
heatMapEUA.save('./heat-map-eua.html')
heatMapEUA
```





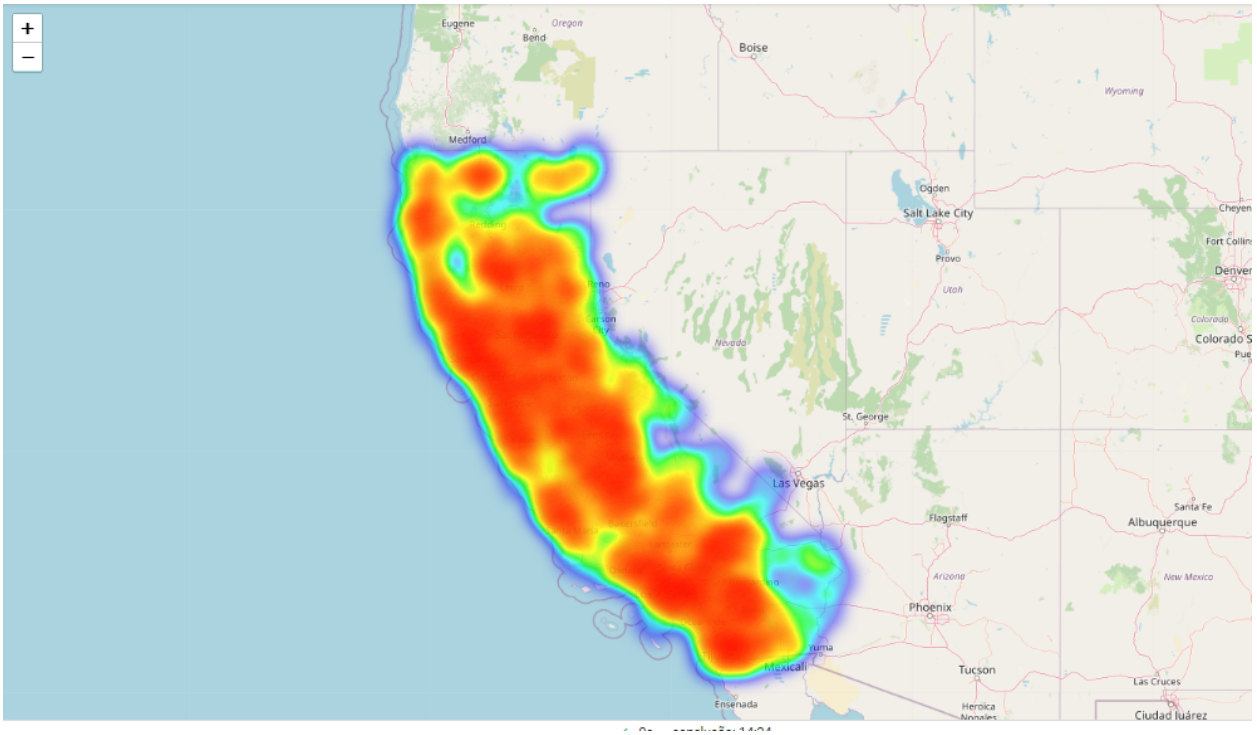
Mapa de pontos dos EUA

```
markerMapEUA = folium.Map(location=[39, -92], zoom_start = 3, prefer_canvas=True)

for coor in plot_data_eua:
    folium.CircleMarker(location=[ coor[0], coor[1]], popup=f'{coor[2]} reporte(s)', radius=2, fill_color='red').add_to(markerMapEUA)

markerMapEUA.save('./circle-map-eua.html')
markerMapEUA
```



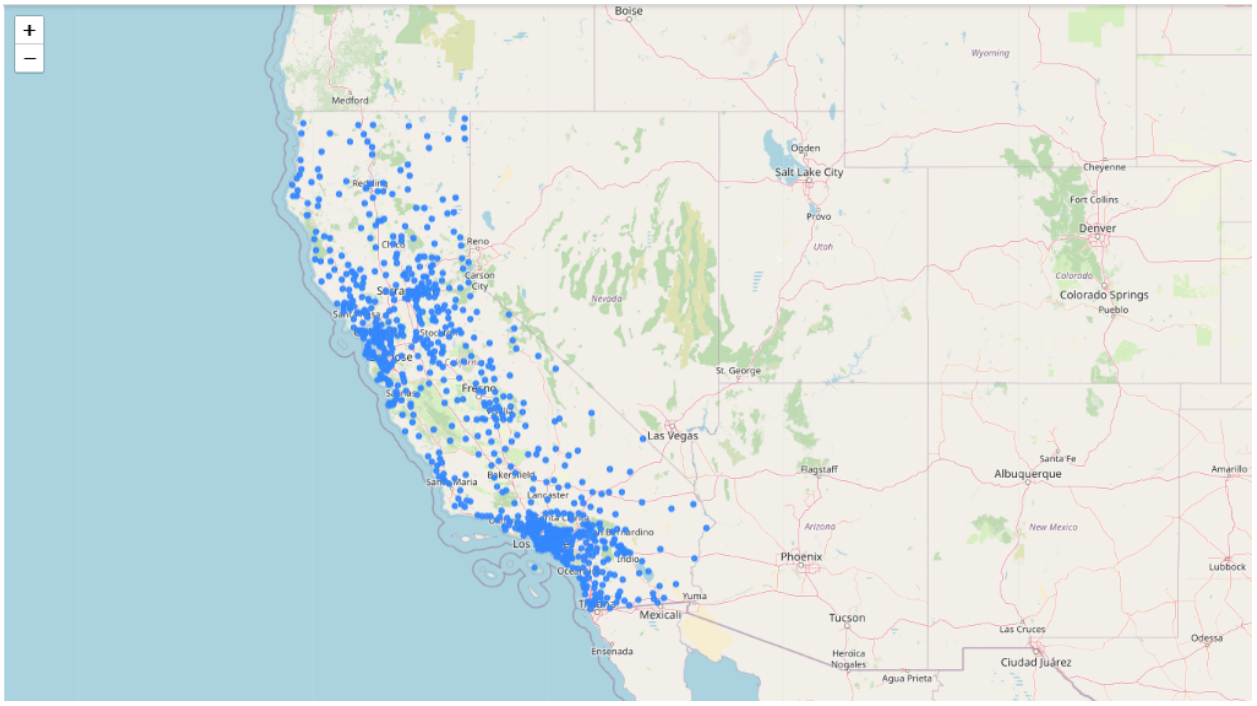
Mapa de Calor da Califórnia

```
heatMapCali = folium.Map(location=[39, -122], zoom_start = 5, prefer_canvas=True)

HeatMap(plot_data_cali).add_to(heatMapCali)
heatMapCali.save('./heat-map-cali.html')
heatMapCali
```



Mapa de Pontos da Califórnia

```
markerMapCali = folium.Map(location=[39, -122], zoom_start = 5, prefer_canvas=True)

for coor in plot_data_cali:
      folium.CircleMarker(location=[ coor[0], coor[1]], popup=f'{coor[2]} reporte(s)', radius=2, fill_color='red').add_to(markerMapCali)

markerMapCali.save('./circle-map-cali.html')
markerMapCali
```



Mapa com GMaps

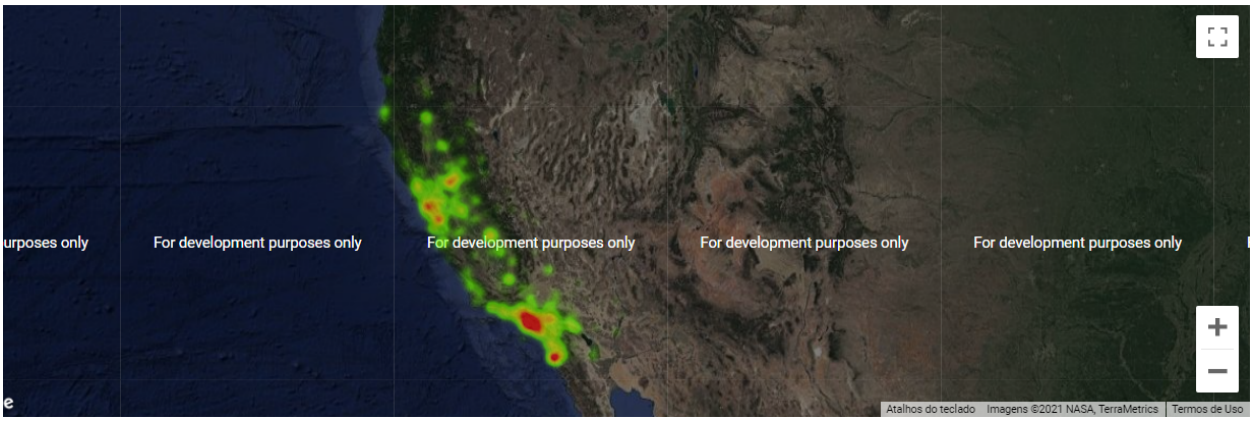
```
gmaps.configure(api_key='AIzaSyDAk3dPCLrZaaU9HAguqxl6iL7cBJul20s')
```

Mapa de Calor dos EUA

```
figEUA = gmaps.figure(map_type='SATELLITE')
figEUA.add_layer(gmaps.heatmap_layer(coordinates[['lat',        'long']],
weights=coordinates['views']))
```

Mapa de Calor da Califórnia

```
figCali = gmaps.figure(map_type='SATELLITE')
figCali.add_layer(gmaps.heatmap_layer(cali_df[['lat',           'long']],
weights=cali_df['views']))

#save
embed_minimal_html('./gmaps-cali.html', views=[figCali])


#save
embed_minimal_html('./gmaps-eua.html', views=[figEUA])
```



Sprint 2.3 - Exploração dos Dados com SQL

```
import pandas as pd
url                                                                    =
'https://raw.githubusercontent.com/infocbra/pratica-integrada-cd-e-am-2021
-1-g1-ghjp/master/1_sprint/ovinis_data.csv?token=AG2WN5KNZSJ3F2RYPKRQRL3BC
LLMU'
df = pd.read_csv(url,sep=',')
```

Carregando o arquivo OVNIS.csv em um dataframe

```
df.drop(df.columns[0], axis=1, inplace=True)
df.head()
```

	data	cidade	estado	formato	duracao	resumo	postado
0	9/30/97 22:00	Madison	WI	Light	5 minutes	Strange light inside Lake Monona	3/2/04
1	9/30/97 20:00	Nova Scotia (Canada)	NS	Light	8-10 seconds.	Single light resembling a star, but moving spu...	10/30/06
2	9/28/97 23:15	San Francisco	CA	Triangle	12-15s	flying-wing shape outlined by 12-14 lights. Ap...	7/5/99
3	9/27/97 23:00	Egan	SD	Other	30 minutes	The Weirdest Thing I Have Ever Seen	2/22/05
4	9/27/97 05:00	Crestwood	KY	Disk	15 minutes	A big disk with red and green lights on the ri...	8/5/01

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100157 entries, 0 to 100156
Data columns (total 7 columns):
 #   Column   Non-Null Count   Dtype 
---  ------   --------------   ----- 
 0   data     100157 non-null  object
 1   cidade   99959 non-null   object
 2   estado   92931 non-null   object
 3   formato  98130 non-null   object
 4   duracao  97156 non-null   object
 5   resumo   100137 non-null  object
 6   postado  100157 non-null  object
dtypes: object(7)
memory usage: 5.3+ MB
```

Removendo registros que tenham valores vazios (None, Unknown, …) para City, State e Shape

```
filtro = ['none','None','Unknown','unknown','nan']

selecao   =   (df['cidade'].isnull())   |   (df['cidade'].isin(filtro))   |
(df['estado'].isnull())        |        (df['estado'].isin(filtro))        |
(df['formato'].isnull()) | (df['formato'].isin(filtro))

df1 = df[~selecao]

df1.dropna(how='all')

df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 84384 entries, 0 to 100156
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   data     84384 non-null  object
 1   cidade   84384 non-null  object
 2   estado   84384 non-null  object
 3   formato  84384 non-null  object
 4   duracao  82372 non-null  object
 5   resumo   84376 non-null  object
 6   postado  84384 non-null  object
dtypes: object(7)
memory usage: 5.2+ MB
```

Manter somente registros referentes aos 51 estados dos EUA

```
url2                                                                =
'https://raw.githubusercontent.com/oliveirafhm/data_science/master/5_pipel
ine_dados/usa_states.csv'
usa_states = pd.read_csv(url2,sep=',')

filtro = usa_states['Abbreviation']
```

```
selecao = df1['estado'].isin(filtro)
df2 = df1[selecao].reset_index(drop=True)
df2
```

	data	cidade	estado	formato	duracao	resumo	postado
0	9/30/97 22:00	Madison	WI	Light	5 minutes	Strange light inside Lake Monona	3/2/04
1	9/28/97 23:15	San Francisco	CA	Triangle	12-15s	flying-wing shape outlined by 12-14 lights. Ap...	7/5/99
2	9/27/97 23:00	Egan	SD	Other	30 minutes	The Weirdest Thing I Have Ever Seen	2/22/05
3	9/27/97 05:00	Crestwood	KY	Disk	15 minutes	A big disk with red and green lights on the ri...	8/5/01
4	9/25/97 22:00	Clearfield	UT	Triangle	60-90 seconds	We observed a low flying craft (aprox.100yards...	1/28/99
...	...	...	...	...	...	...	...
80486	8/1/17 06:15	Columbus (North)	GA	Fireball	3 seconds	Green streak growing in size moving from west ...	8/4/17
80487	8/1/17 02:45	Corcoran	MN	Light	Still going	Small light south west of Minneapolis maneuver...	8/4/17
80488	8/1/17 02:00	Moreno Valley	CA	Other	10 seconds	I was looking out the front windshield and loo...	8/4/17
80489	8/1/17 01:00	Bradenton	FL	Other	<20 seconds	I was walking my dog about 1am on August 1, 20...	5/9/19
80490	8/1/17	Laurel	MD	Other	NaN	It was an alien project level 1 federal ran on...	6/25/20

80491 rows × 7 columns

Removendo variáveis irrelevantes para análise (Duration, Summary e Posted)

```
df3 = df2.drop(df2.columns[4:], axis=1)
df3
```

	data	cidade	estado	formato
0	9/30/97 22:00	Madison	WI	Light
1	9/28/97 23:15	San Francisco	CA	Triangle
2	9/27/97 23:00	Egan	SD	Other
3	9/27/97 05:00	Crestwood	KY	Disk
4	9/25/97 22:00	Clearfield	UT	Triangle
...	...	...	...	...
80486	8/1/17 06:15	Columbus (North)	GA	Fireball
80487	8/1/17 02:45	Corcoran	MN	Light
80488	8/1/17 02:00	Moreno Valley	CA	Other
80489	8/1/17 01:00	Bradenton	FL	Other
80490	8/1/17	Laurel	MD	Other

80491 rows × 4 columns

Mantendo somente os registros de Shapes mais populares (com mais de 1000 ocorrências)

```
selecao = df3['formato'].value_counts(sort=True) >= 1000
selecao = selecao[selecao].index
df4 = df3[df3['formato'].isin(selecao)].reset_index()
df4
```

	index	data	cidade	estado	formato
0	0	9/30/97 22:00	Madison	WI	Light
1	1	9/28/97 23:15	San Francisco	CA	Triangle
2	2	9/27/97 23:00	Egan	SD	Other
3	3	9/27/97 05:00	Crestwood	KY	Disk
4	4	9/25/97 22:00	Clearfield	UT	Triangle
...	...	...	...	...	...
78212	80486	8/1/17 06:15	Columbus (North)	GA	Fireball
78213	80487	8/1/17 02:45	Corcoran	MN	Light
78214	80488	8/1/17 02:00	Moreno Valley	CA	Other
78215	80489	8/1/17 01:00	Bradenton	FL	Other
78216	80490	8/1/17	Laurel	MD	Other

78217 rows × 5 columns

```
df5 = df4.drop(df4.columns[0], axis=1)
df5
```

	data	cidade	estado	formato
0	9/30/97 22:00	Madison	WI	Light
1	9/28/97 23:15	San Francisco	CA	Triangle
2	9/27/97 23:00	Egan	SD	Other
3	9/27/97 05:00	Crestwood	KY	Disk
4	9/25/97 22:00	Clearfield	UT	Triangle
...	...	...	...	...
78212	8/1/17 06:15	Columbus (North)	GA	Fireball
78213	8/1/17 02:45	Corcoran	MN	Light
78214	8/1/17 02:00	Moreno Valley	CA	Other
78215	8/1/17 01:00	Bradenton	FL	Other
78216	8/1/17	Laurel	MD	Other

78217 rows × 4 columns

Salvando o dataframe em um arquivo CSV com o nome “df_OVNI_limpo”

```
from google.colab import drive
drive.mount('drive')
df5.to_csv('df_OVNI_limpo.csv')
!cp df_OVNI_limpo.csv "drive/My Drive/"
```

Sprint 2.4 - Acréscimo de Variáveis

Sprint 2.4 - Acréscimo de Variáveis

```
import pandas as pd
url = 'https://raw.githubusercontent.com/infocbra/pratica-integrada-cd-e-am-2021-1-g1-ghjp/master/2_sprint/df_OVNI_limpo.csv?token=AG2WN5PEBBBSDKOT3LMM3I3BEQCSK'
df = pd.read_csv(url,sep=';',encoding = "ISO-8859-1")
```

1 -Aspecto do dataframe resultado da atividade 2.3;

```
df.drop(df.columns[0], axis=1, inplace=True)
df.head()
```

	data	cidade	estado	formato
0	9/30/97 22:00	Madison	WI	Light
1	9/28/97 23:15	San Francisco	CA	Triangle
2	9/27/97 23:00	Egan	SD	Other
3	9/27/97 05:00	Crestwood	KY	Disk
4	9/25/97 22:00	Clearfield	UT	Triangle

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 78217 entries, 0 to 78216
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   data     78217 non-null  object
 1   cidade   78217 non-null  object
 2   estado   78217 non-null  object
 3   formato  78217 non-null  object
dtypes: object(4)
```

2 - Dividir o conteúdo da coluna Date / Time em duas novas colunas no mesmo dataframe e deletar a coluna Date / Time ;

```
df1 = df


df1['visualizacao_data'] = pd.to_datetime(df['data']).dt.date
df1['visualizacao_horario'] = pd.to_datetime(df['data']).dt.time
```

```
df1.drop(df.columns[0], axis=1, inplace=True)
df1.head()
```

	cidade	estado	formato	visualizacao_data	visualizacao_horario
0	Madison	WI	Light	1997-09-30	22:00:00
1	San Francisco	CA	Triangle	1997-09-28	23:15:00
2	Egan	SD	Other	1997-09-27	23:00:00
3	Crestwood	KY	Disk	1997-09-27	05:00:00
4	Clearfield	UT	Triangle	1997-09-25	22:00:00

3 - Fazer o mesmo procedimento para dias da semana. Será que existe um dia da semana com mais ocorrências de relatórios para OVNIs? Para descobrir isso, você deve criar uma nova coluna chamada weekdays.

```
import datetime
df2 = df1
df2['weekdays'] = pd.to_datetime(df2['visualizacao_data'], errors='coerce').dt.day_name()
df2
```

	cidade	estado	...	visualizacao_horario	weekdays
0	Madison	WI	...	22:00:00	Tuesday
1	San Francisco	CA	...	23:15:00	Sunday
2	Egan	SD	...	23:00:00	Saturday
3	Crestwood	KY	...	05:00:00	Saturday
4	Clearfield	UT	...	22:00:00	Thursday
...	...	...	...	...	...
78212	Columbus (North)	GA	...	06:15:00	Tuesday
78213	Corcoran	MN	...	02:45:00	Tuesday
78214	Moreno Valley	CA	...	02:00:00	Tuesday
78215	Bradenton	FL	...	01:00:00	Tuesday
78216	Laurel	MD	...	00:00:00	Tuesday

[78217 rows x 6 columns]

O dia com maior número de ocorrências é o Sábado

```
df2['weekdays'].value_counts(sort=True)
```

```
Saturday     14545
Friday       11395
Sunday       11304
Wednesday    10612
Thursday     10603
Tuesday      10163
Monday        9595
Name: weekdays, dtype: int64
```

4 - Separar as variáveis mês (Month) e dia (Day). Desse modo, será possível refinar as pesquisas.;

```
df3 = df2
df3['visualizacao_dia'] = pd.to_datetime(df2['visualizacao_data']).dt.day
df3['visualizacao_mes'] = pd.to_datetime(df2['visualizacao_data']).dt.month
df3
```

	cidade	estado	...	visualizacao_dia	visualizacao_mes
0	Madison	WI	...	30	9
1	San Francisco	CA	...	28	9
2	Egan	SD	...	27	9
3	Crestwood	KY	...	27	9
4	Clearfield	UT	...	25	9
...	...	...	...	...	...
78212	Columbus (North)	GA	...	1	8
78213	Corcoran	MN	...	1	8
78214	Moreno Valley	CA	...	1	8
78215	Bradenton	FL	...	1	8
78216	Laurel	MD	...	1	8

[78217 rows x 8 columns]

5 - Por fim, salvar o dataframe resultante em um arquivo .csv com o nome: 'df_OVNI_preparado'.

```
from google.colab import drive
drive.mount('drive')
df3.to_csv('df_OVNI_preparado.csv')
!cp df_OVNI_preparado.csv "drive/My Drive/Colab Notebooks/projeto_integrado/"
```

```
Drive already mounted at drive; to attempt to forcibly remount, call drive.mount("drive", force_remount=True).
```

4. Considerações finais

Entendemos que a visualização e o entendimento dos dados foi melhor aproveitado através da construção de gráficos e mapas, entretanto, a equipe encontrou muitas dificuldades na etapa de limpeza e preparação dos dados para poder assim plotá-los em gráficos e mapas.

O desafio consistiu em deixar os dados o mais limpo possível, retirando todas as informações desnecessárias e os possíveis registros equivocados.

Havia uma proporção de conhecimento desigual entre os integrantes da equipe sobre as ferramentas e bibliotecas utilizadas, o que ocasionou um gasto maior de tempo para a conclusão da sprint.

De modo geral, a experiência no desenvolvimento das soluções agregou conhecimento à equipe.

Referências

Deserto da Califórnia atrai curiosos em busca de objetos voadores não identificados. G1 Portal de notícias. Disponível em: <https://g1.globo.com/globo-reporter/noticia/2021/06/19/deserto-da-california-atrai-curiosos-em-busca-de-objetos-voadores-nao-identificados.ghtml>. Acesso em: 13 de agosto de 2021.