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.

2. Descrição do problema

Severo (2017) afirma que a Ufologia é considerada um ramo da ciência não exata, seja pelo seu caráter, por vezes, especulativo, seja pela dificuldade de obtenção de dados provenientes de fontes confiáveis, estando, na verdade, muito mais próxima do ramo de investigação especulativa do que de uma ciência de fato.

Mesmo com questionamentos acerca da falta de responsabilidade quanto às técnicas científicas adotadas e metodologias desconexas, o tema ufologia gera um grande arcabouço de dados como, por exemplo, os encontrados no site Nuforc - National UFO Reporting Center.

Ainda que a temática possa ser considerada especulativa ela ainda é uma fonte de dados sobre um tema de relevante interesse para diversos ramos da sociedade. Dessa forma, o problema desta etapa é explorar a maior quantidade possível de dados interessantes, sendo eles:

1. Saber a quantidade de linhas, observações ou variáveis que foram coletadas.
2. Quantos relatos ocorreram por estado em ordem decrescente?
3. Remover possíveis campos vazios (sem estado).
4. Limitar a análise aos estados dos Estados Unidos.
5. Consulta por cidades, com o objetivo de saber quais contêm o maior número de relatos (cidades que apresentem ao menos 10 relatos).
6. Com o dado anterior, responder a seguinte pergunta: por que será que essa é a cidade que possui mais relatos?
7. Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidades que possuam um número superior a 10 relatórios. Enfatizar a cidade, a quantidade de relatos e formato do objeto não identificado.

3. Desenvolvimento

Após a coleta os dados foram submetidos a limpeza e a análise exploratória com o intuito de obter conhecimento sobre os mesmos e assim coletar fatos interessantes que apoiarão as próximas fases desta pesquisa, além de compreender quais conhecimentos podem ser evidenciados, através da exploração dos dados da plataforma Nuforc, utilizando para isso paradigmas inseridos nas áreas de aprendizagem de máquina e ciência de dados.

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

- Requests: permite acessar uma URL e baixar o HTML disponível nela;
- BeautifulSoup: permite a leitura o HTML de maneira estruturada em objetos que podem ser inspecionados pelo PYTHON;
- Pandas: permite a manipulação e análise de dados.

Após a raspagem dos dados, realizou-se análise exploratória a fim de responder aos problemas definidos nesta etapa do projeto.

3.1 Código implementado

Classe scrapping.py

```
from typing import List
import datetime
from dateutil.relativedelta import relativedelta

from pandas import DataFrame
import requests
from bs4 import BeautifulSoup
import time


class Scrapping:

    @staticmethod
```

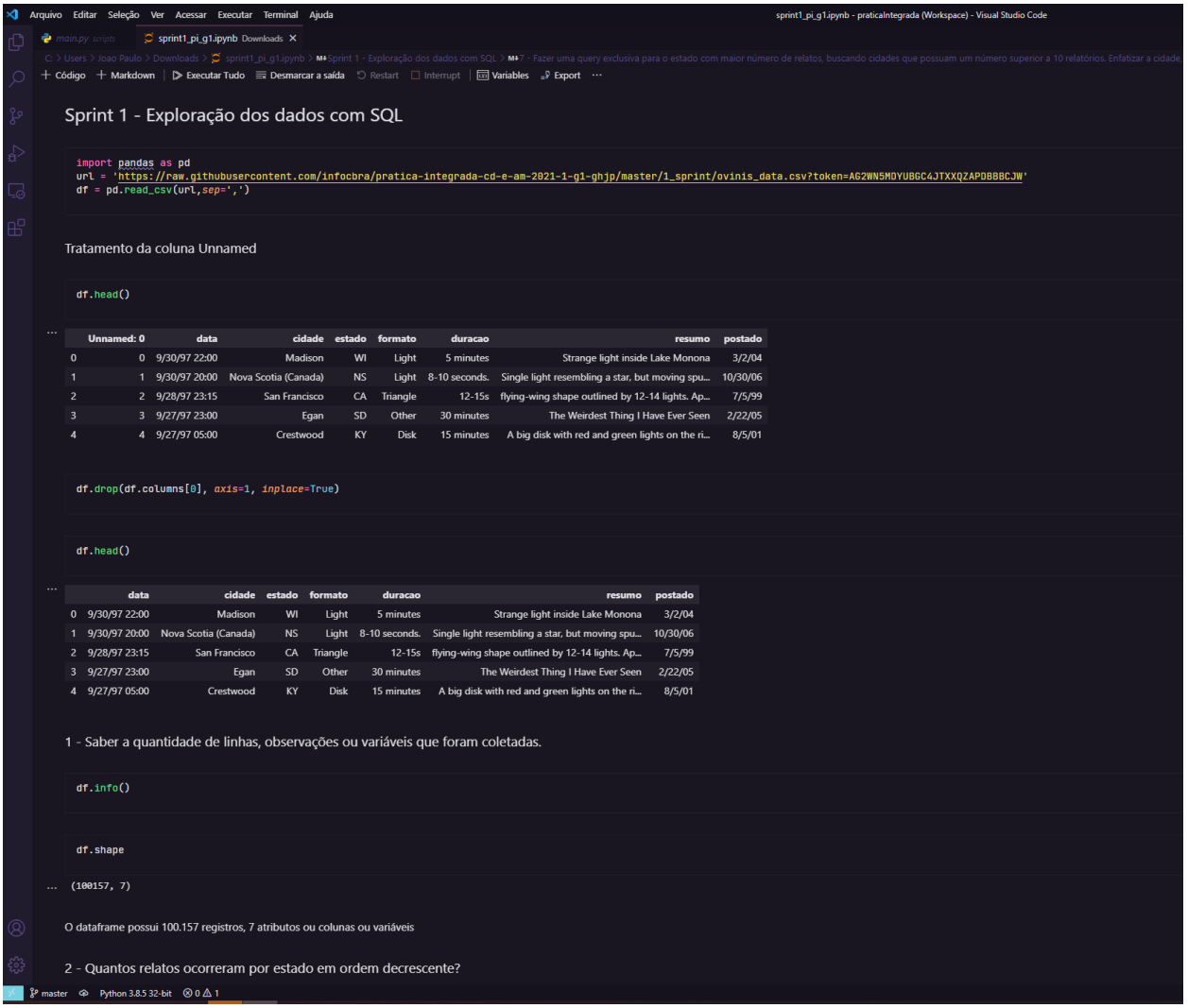
```
            def generate_urls(starting_year: int, starting_month: int,
ending_year: int, ending_month: int) -> List[int]:
                '''
                      Retorna uma lista de urls válidas do site de acordo
com o intervalo das datas inseridas.
                    padrão da data: yyyymm
                    padrão de url: http://www.nuforc.org/webreports/ndxe +
yyyymm + .html
                '''
                urls = []

                            starting_date   =   datetime.date(starting_year,
starting_month, 1)
                ending_date = datetime.date(ending_year, ending_month, 1)

                while starting_date <= ending_date:

urls.append(starting_date.strftime('http://www.nuforc.org/webreports/ndx
e%Y%m.html'))
                    starting_date += relativedelta(months=1)

                return urls

            @staticmethod
            def get_data(urls: List[str]) -> DataFrame:
                '''
                    Retorna um dataframe contento os dados coletados
                    das urls fornecidas.
                '''
                     df  =  DataFrame(columns=['data',  'cidade',  'estado',
'formato', 'duracao', 'resumo', 'postado'])

                for url in urls:
                    html_page = requests.get(url)
                    soup = BeautifulSoup(html_page.text, "html.parser")

                    table = soup.find('table').find_all('tr')

                    collection = []
                    for row in table[1:]:
                        cells = row.find_all('td')
                        collection.append(
                            {
                                'data': cells[0].text,
                                'cidade': cells[1].text,
                                'estado': cells[2].text,
```

```
                            'formato': cells[3].text,
                            'duracao': cells[4].text,
                            'resumo': cells[5].text,
                            'postado': cells[6].text
                        }
                    )

                    df_row = DataFrame(data=collection)
                df = df.append(df_row, ignore_index=True)
                time.sleep(2)

            return df


        @staticmethod
                def generate_df_from_dataset(starting_year: int,
starting_month: int, ending_year: int, ending_month: int) -> DataFrame:
                    urls = Scrapping.generate_urls(starting_year,
starting_month, ending_year, ending_month)
            return Scrapping.get_data(urls)


        @staticmethod
                def generate_csv_from_dataset(starting_year: int,
starting_month: int, ending_year: int, ending_month: int, path: str) ->
DataFrame:
                    urls = Scrapping.generate_urls(starting_year,
starting_month, ending_year, ending_month)
            df = Scrapping.get_data(urls)
            df.to_csv(path)
```

Classe main.py

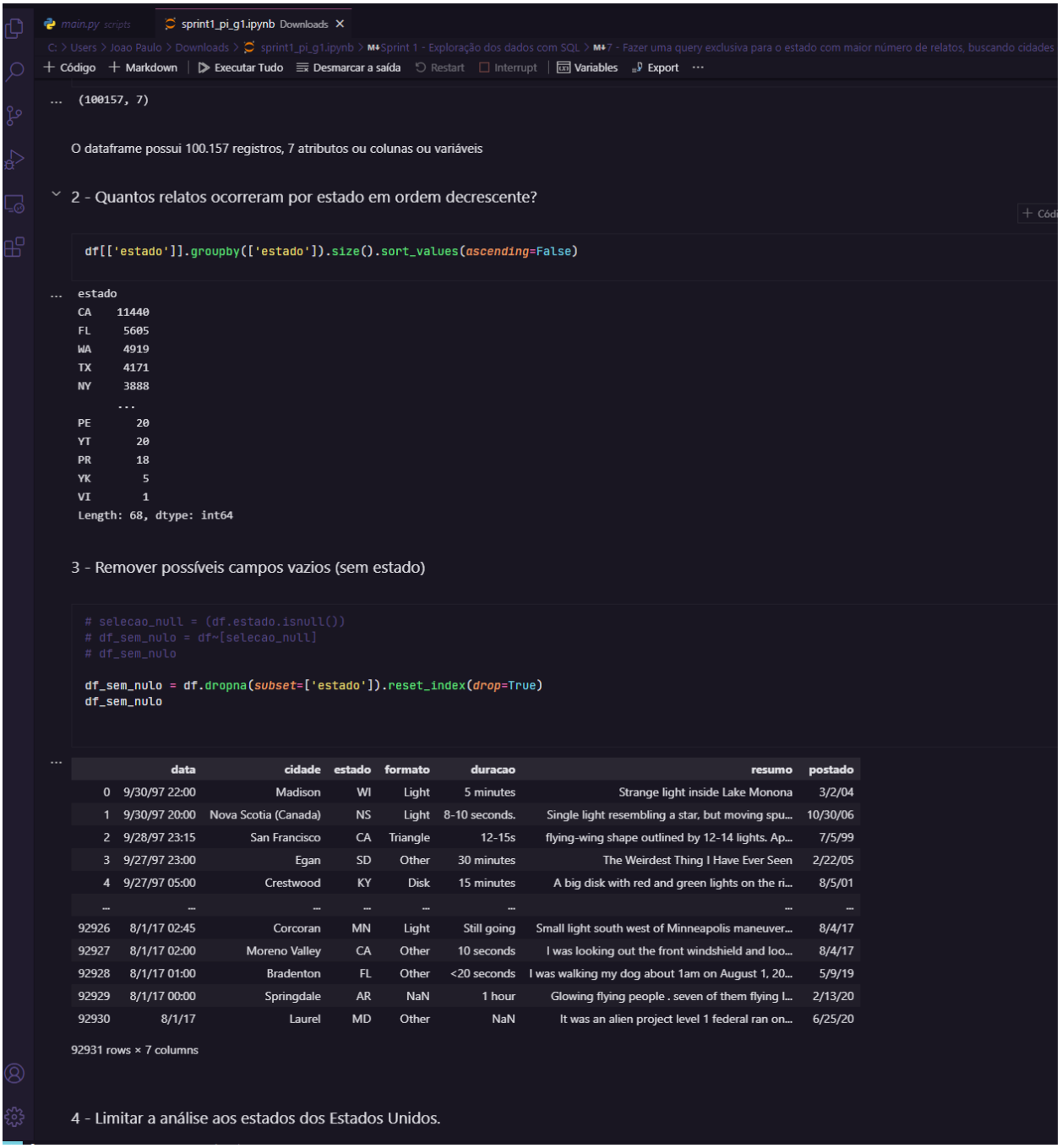
```
from scrapping import Scrapping

Scrapping.generate_csv_from_data(1997, 9, 2017, 8, '../ovinis_data.csv')
```

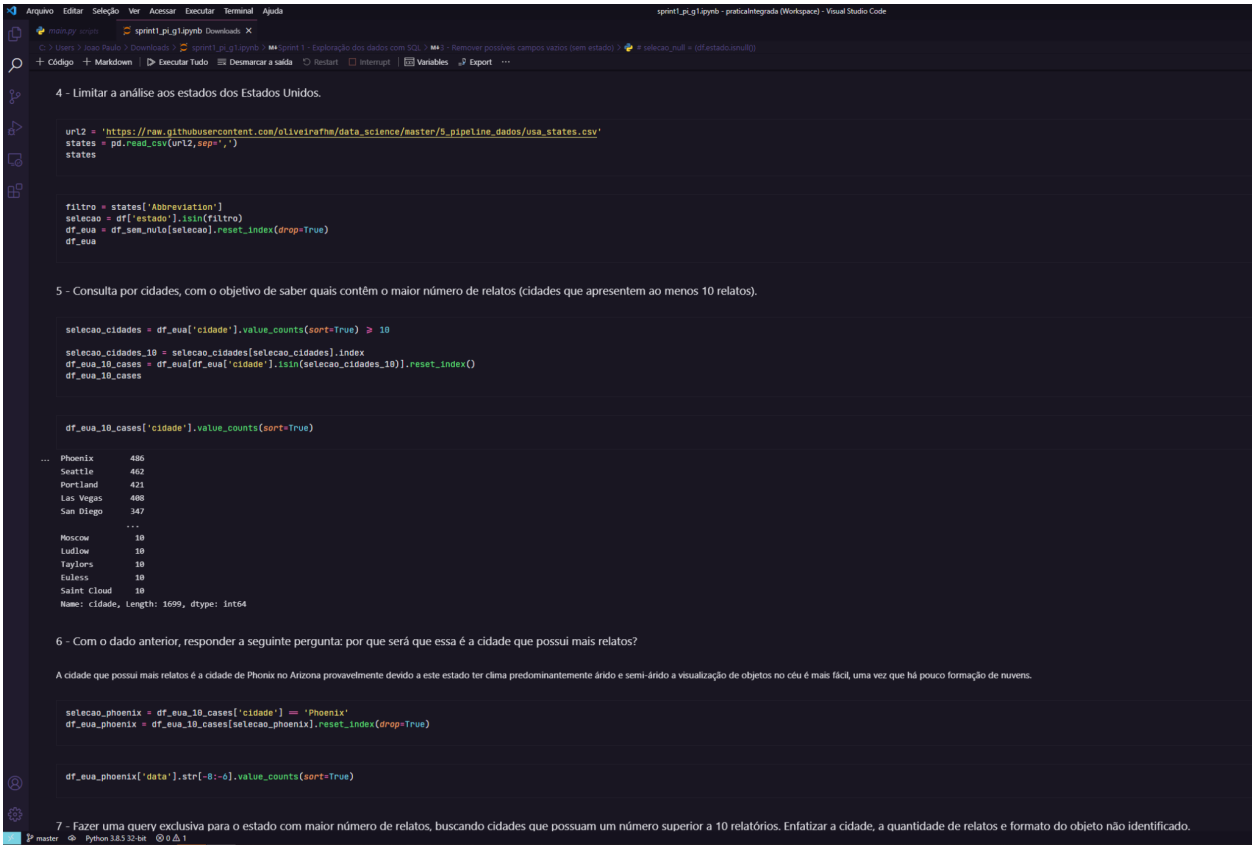
Análise Exploratória

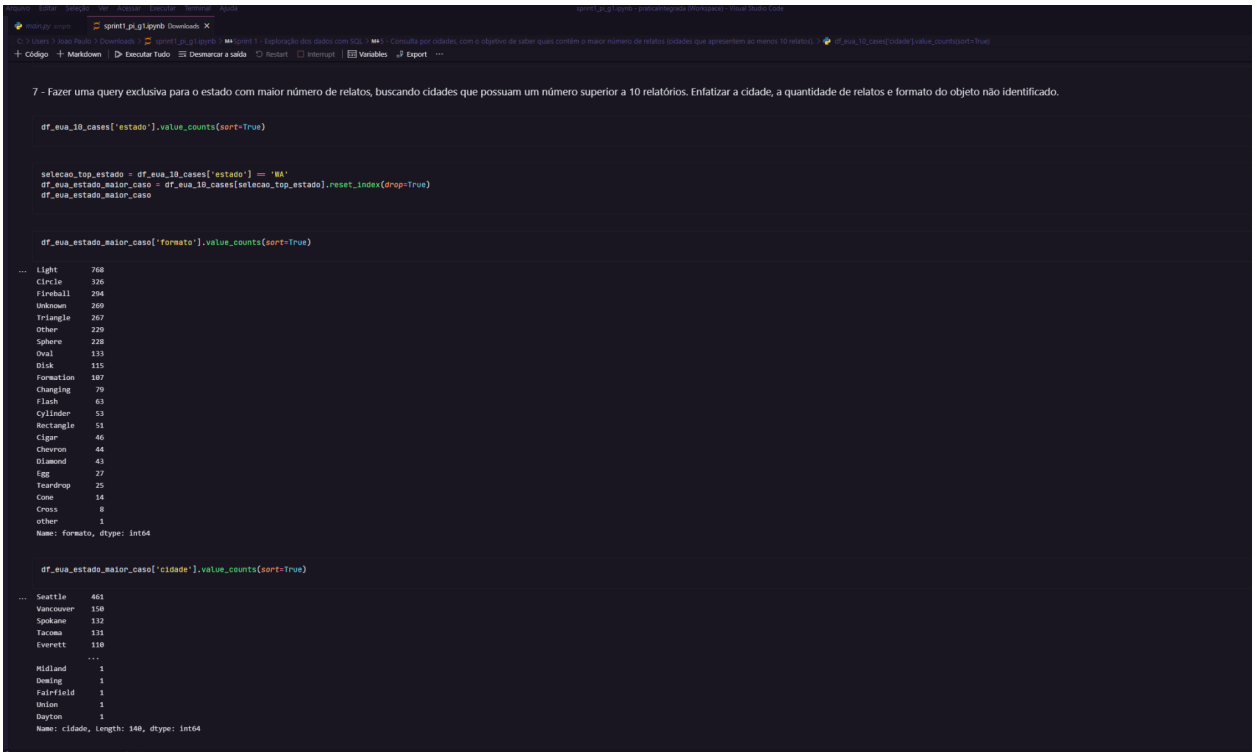


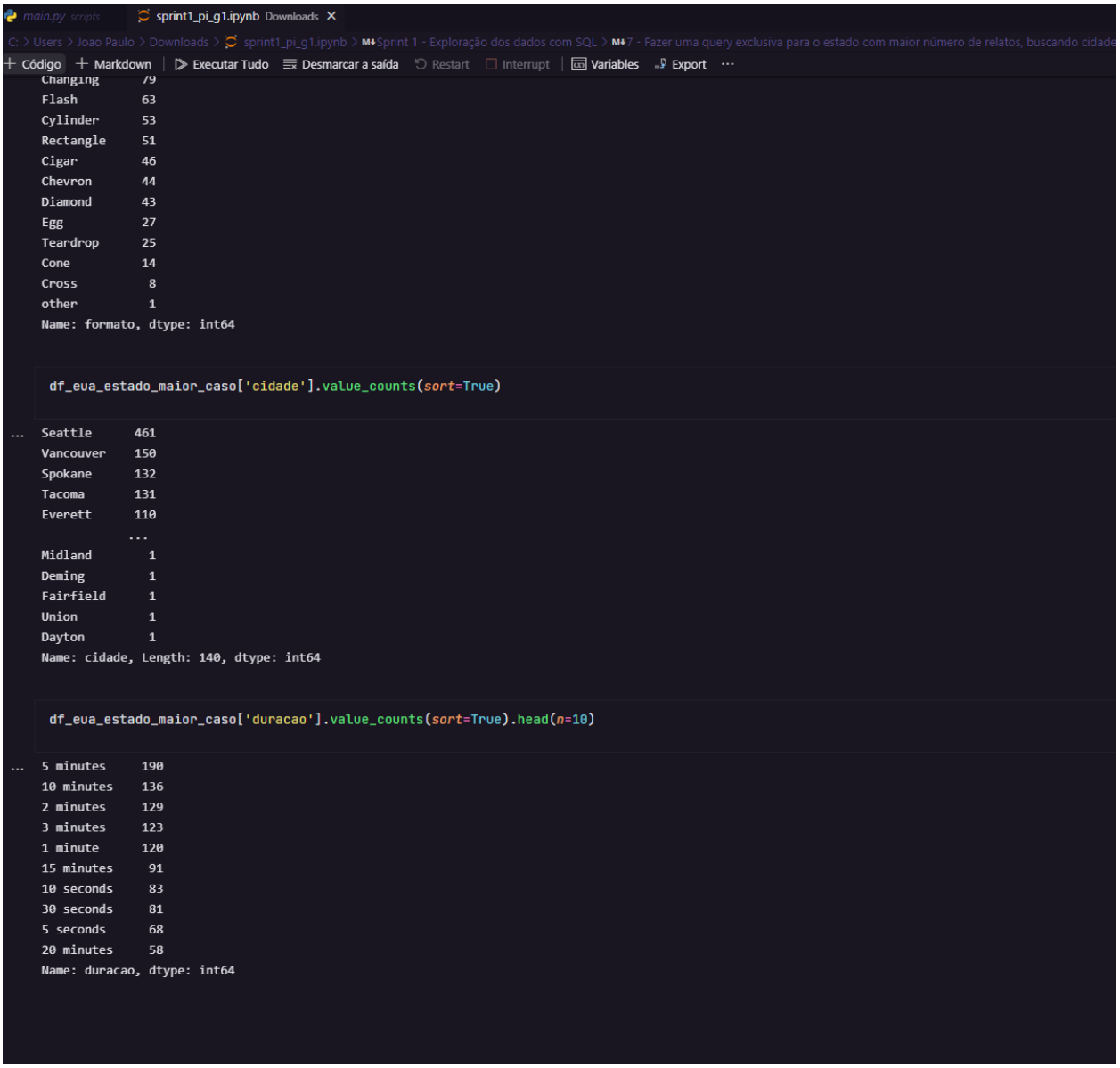
	Unnamed: 0	data	cidade	estado	formato	duracao	resumo	postado
0	0	9/30/97 22:00	Madison	WI	Light	5 minutes	Strange light inside Lake Monona	3/2/04
1	1	9/30/97 20:00	Nova Scotia (Canada)	NS	Light	8-10 seconds.	Single light resembling a star, but moving spu...	10/30/06
2	2	9/28/97 23:15	San Francisco	CA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

	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


main.py scripts
sprint1_pi_g1.ipynb Downloads
C: > Users > Joao Paulo > Downloads > sprint1_pi_g1.ipynb > Sprint 1 - Exploração dos dados com SQL > 7 - Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidades
+ Código + Markdown Executar Tudo Desmarcar a saída Restart Interrupt Variables Export
(100157, 7)
O dataframe possui 100.157 registros, 7 atributos ou colunas ou variáveis
2 - Quantos relatos ocorreram por estado em ordem decrescente?
df[['estado']].groupby(['estado']).size().sort_values(ascending=False)
estado
CA 11440
FL 5605
WA 4919
TX 4171
NY 3888
...
PE 20
YT 20
PR 18
YK 5
VI 1
Length: 68, dtype: int64
3 - Remover possíveis campos vazios (sem estado)
# selecao_null = (df.estado.isnull())
# df_sem_nulo = df~[selecao_null]
# df_sem_nulo
df_sem_nulo = df.dropna(subset=['estado']).reset_index(drop=True)
df_sem_nulo
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
92926 8/1/17 02:45 Corcoran MN Light Still going Small light south west of Minneapolis maneuver... 8/4/17
92927 8/1/17 02:00 Moreno Valley CA Other 10 seconds I was looking out the front windshield and loo... 8/4/17
92928 8/1/17 01:00 Bradenton FL Other <20 seconds I was walking my dog about 1am on August 1, 20... 5/9/19
92929 8/1/17 00:00 Springdale AR NaN 1 hour Glowing flying people . seven of them flying l... 2/13/20
92930 8/1/17 Laurel MD Other NaN It was an alien project level 1 federal ran on... 6/25/20
92931 rows × 7 columns
4 - Limitar a análise aos estados dos Estados Unidos.


Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
sprint1_pi_g1.ipynb - praticaIntegrada (Workspace) - Visual Studio Code
sprint1_pi_g1.ipynb
+ Código + Markdown Executar Tudo Desmarcar a saída Restart Interrupt Variables Export
4 - Limitar a análise aos estados dos Estados Unidos.
url2 = 'https://raw.githubusercontent.com/oliveirafhm/data_science/master/5_pipeline_dados/usa_states.csv'
states = pd.read_csv(url2,sep=',')
states
filtro = states['Abbreviation']
selecao = df['estado'].isin(filtro)
df_eua = df_sem_nulo[selecao].reset_index(drop=True)
df_eua
5 - Consulta por cidades, com o objetivo de saber quais contêm o maior número de relatos (cidades que apresentem ao menos 10 relatos).
selecao_cidades = df_eua['cidade'].value_counts(sort=True) ≥ 10
selecao_cidades_10 = selecao_cidades[selecao_cidades].index
df_eua_10_cases = df_eua[df_eua['cidade'].isin(selecao_cidades_10)].reset_index()
df_eua_10_cases
df_eua_10_cases['cidade'].value_counts(sort=True)
Phoenix 486
Seattle 462
Portland 421
Las Vegas 408
San Diego 347
Moscow 10
Ludlow 10
Taylors 10
Euless 10
Saint Cloud 10
Name: cidade, Length: 1699, dtype: int64
6 - Com o dado anterior, responder a seguinte pergunta: por que será que essa é a cidade que possui mais relatos?
A cidade que possui mais relatos é a cidade de Phonix no Arizona provavelmente devido a este estado ter clima predominantemente árido e semi-árido a visualização de objetos no céu é mais fácil, uma vez que há pouco formação de nuvens.
selecao_phoenix = df_eua_10_cases['cidade'] == 'Phoenix'
df_eua_phoenix = df_eua_10_cases[selecao_phoenix].reset_index(drop=True)
df_eua_phoenix['data'].str[-8:-6].value_counts(sort=True)
7 - Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidades que possuam um número superior a 10 relatórios. Enfatizar a cidade, a quantidade de relatos e formato do objeto não identificado.
master Python 3.8.5 32-bit 0 1


sprint1_pi_g1.ipynb
+ Código + Markdown Executar Tudo Desmarcar a saída Restart Interrupt Variables Export
7 - Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidades que possuam um número superior a 10 relatórios. Enfatizar a cidade, a quantidade de relatos e formato do objeto não identificado.
df_eua_10_cases['estado'].value_counts(sort=True)
selecao_top_estado = df_eua_10_cases['estado'] == 'WA'
df_eua_estado_maior_caso = df_eua_10_cases[selecao_top_estado].reset_index(drop=True)
df_eua_estado_maior_caso
df_eua_estado_maior_caso['formato'].value_counts(sort=True)
Light 768
Circle 326
Fireball 294
Unknown 269
Triangle 267
Other 229
Sphere 228
Oval 133
Disk 115
Formation 107
Changing 79
Flash 63
Cylinder 53
Rectangle 51
Cigar 46
Chevron 44
Diamond 43
Egg 27
Teardrop 25
Cone 14
Cross 8
other 1
Name: formato, dtype: int64
df_eua_estado_maior_caso['cidade'].value_counts(sort=True)
Seattle 461
Vancouver 150
Spokane 132
Tacoma 131
Everett 110
Midland 1
Deming 1
Fairfield 1
Union 1
Dayton 1
Name: cidade, Length: 140, dtype: int64


main.py scripts
sprint1_pi_g1.ipynb Downloads
C: > Users > Joao Paulo > Downloads > sprint1_pi_g1.ipynb > Sprint 1 - Exploração dos dados com SQL > 7 - Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidade
+ Código + Markdown Executar Tudo Desmarcar a saída Restart Interrupt Variables Export ...
Changing 79
Flash 63
Cylinder 53
Rectangle 51
Cigar 46
Chevron 44
Diamond 43
Egg 27
Teardrop 25
Cone 14
Cross 8
other 1
Name: formato, dtype: int64
df_eua_estado_maior_caso['cidade'].value_counts(sort=True)
... Seattle 461
Vancouver 150
Spokane 132
Tacoma 131
Everett 110
...
Midland 1
Deming 1
Fairfield 1
Union 1
Dayton 1
Name: cidade, Length: 140, dtype: int64
df_eua_estado_maior_caso['duracao'].value_counts(sort=True).head(n=10)
... 5 minutes 190
10 minutes 136
2 minutes 129
3 minutes 123
1 minute 120
15 minutes 91
10 seconds 83
30 seconds 81
5 seconds 68
20 minutes 58
Name: duracao, dtype: int64

4. Considerações finais

Entendemos que o desafio da raspagem dos dados apresentou uma dificuldade maior que a equipe possuía, sendo que apenas um integrante tinha maior domínio e conseguiu resolver o problema.

A análise dos dados apresentou uma dificuldade moderada, mas a equipe conseguiu pesquisar e pensar em soluções para a exploração dos dados.

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

Referências

Aqui vocês podem colocar quaisquer referências externas que tenham utilizado (Sugiro colocar na ABNT, porque a falta de padrão dificulta a leitura).

SEVERO, Maria Eduarda Porto et al. UFOLOGIA. **ANAIS CONGREGA MIC-ISBN: 978-65-86471-05-2 e ANAIS MIC JR.-ISBN: 978-65-86471-06-9**, n. 12, p. 86, 2017.

Reis, Carlos, and Ubirajara Rodrigues. "Discos voadores: entre a crença e o conhecimento." (2011): 209-223.