Relatório UFO Data – Parte 4

Levi Alves de Freitas Junior

Problema

Realizar uma limpeza no DataFrame OVNI, retirar campos em Branco, Unknown e None, realizar uma filtragem com pandasql para explorar e eliminar os dados. Vamos utilizar a base ovni_data gerada no primeiro relatório.

Etapa inicial criar um novo notebook no colab research. Siga as instruções a seguir e veja as imagens ilustrativas:

1 - Após criar um novo notebook, podemos começar a criar nosso código e inicialmente precisamos importar as bibliotecas pandas e para usar o banco de dados vamos importar o pandasql.

```
#!pip install -U pandasql
import pandas as pd
import pandasql
```

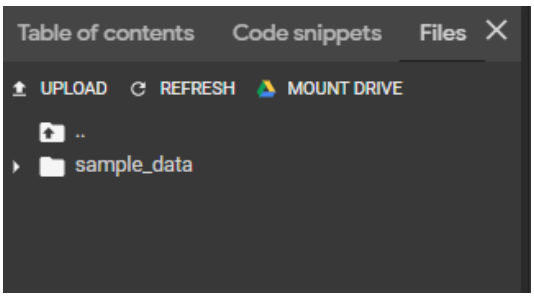
Primeiro selecione o comando - !pip install -U pandasql e aperte CTRL+ SHIFT + ENTER para instalar o pandasql no colab, logo após inicialize a célula no colab para importar o pandas e o pandasql,

Obs. O import do pandasql pode não funcionar antes de utilizar o comando pip para utilizá-lo.

2 – Agora vamos importar a base de dados

Lembre-se de importar a base de dados no colab para podermos chama-la no código, siga as instruções abaixo para colocar a base.

No menu esquerdo selecione Files, e clique em upload, selecione a base, onde você salvou



Aguarde um pouco e sua base estará inserida no colab.

Podemos prosseguir agora e chamar nossa base de dados.

```
df = pd.read_csv('ovni_data.csv')

#Excluindo coluna Indesejada
del df['Unnamed: 0']
```

O comando ‘del’ retiramos a coluna não nomeada pois ela não terá nenhuma utilidade para o projeto.

3 - Remover registros que tenham valores vazios (*None*, *Unknown*, ...) para City, State e Shape.

```
eliminar_campos = """
   SELECT *
   FROM df
   WHERE lower(City) not in ('unknown', 'none', ' ') and
   lower(State) not in ('unknown', 'none', ' ') and
   lower(Shape) not in ('unknown', 'none', ' ')
 """
lista_result_eliminacao = pandasql.sqldf(eliminar_campos)
```

Utilizamos o pandasql para realizar a filtragem e remoção dos dados, selecionamos todos os dados do dataframe DF e na cláusula WHERE colocamos a restrição dos itens a não serem retornados após a pesquisa.

Como resultado recebemos o dataframe a seguir:

	Date / Time	City	State	Shape	Duration	Summary	Posted
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
...	...	...	...	...	...	...	...
84024	8/1/17 14:00	Joliet	IL	Other	2 minutes	The White Cube UFO	7/25/19
84025	8/1/17 06:15	Columbus (North)	GA	Fireball	3 seconds	Green streak growing in size moving from west ...	8/4/17
84026	8/1/17 02:45	Corcoran	MN	Light	Still going	Small light south west of Minneapolis maneuver...	8/4/17
84027	8/1/17 02:00	Moreno Valley	CA	Other	10 seconds	I was looking out the front windshield and loo...	8/4/17
84028	8/1/17 01:00	Bradenton	FL	Other	<20 seconds	I was walking my dog about 1am on August 1, 20...	5/9/19

84029 rows × 7 columns

4 – Manter os registros referentes aos 51 estados dos Estados Unidos.

Para esta etapa baixe o arquivo no GitHub chamado states.csv e logo após importe-o no seu Files do colab.

```
estados = pd.read_csv('states.csv', sep=';')
```

Agora comparamos os estados do arquivo states.csv com os estados de lista_result_eliminacao

```
consulta_estados_validos = '''
  SELECT lista_result_eliminacao.*
  FROM lista_result_eliminacao, estados
  WHERE lista_result_eliminacao.State = estados.Abbreviation
'''

retorno_filtro_usa = pandasql.sqldf(consulta_estados_validos)
```

Este bloco de pandasql retornará um dataframe mais filtrado, vamos observar

	Date / Time	City	State	Shape	Duration	Summary	Posted
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
...	...	...	...	...	...	...	...
80154	8/1/17 14:00	Joliet	IL	Other	2 minutes	The White Cube UFO	7/25/19
80155	8/1/17 06:15	Columbus (North)	GA	Fireball	3 seconds	Green streak growing in size moving from west ...	8/4/17
80156	8/1/17 02:45	Corcoran	MN	Light	Still going	Small light south west of Minneapolis maneuver...	8/4/17
80157	8/1/17 02:00	Moreno Valley	CA	Other	10 seconds	I was looking out the front windshield and loo...	8/4/17
80158	8/1/17 01:00	Bradenton	FL	Other	<20 seconds	I was walking my dog about 1am on August 1, 20...	5/9/19

80159 rows × 7 columns

Observe se seu resultado seja igual à quantidade de linhas que ele retorna que no exemplo acima, no caso, 80159 linhas.

5 – Remover variáveis irrelevantes para análise (Duration, Summary e Posted).

```
excluir_col = retorno_filtro_usa.drop(['Duration', 'Summary', 'Posted'], axis=1)
```

Utilizamos o drop para retirar as colunas Duration, Summary e Posted. O drop é uma biblioteca utilizada no pandas para eliminar as colunas do dataframe, e utilizamos o axis para referenciar o eixo do dataframe, por exemplo (horizontal = 1 ou vertical = 0).

Selecione excluir_col e aperte CTRL + SHIFT + ENTER e seu resultado será o seguinte:

	Date / Time	City	State	Shape
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
...	...	...	...	...
80154	8/1/17 14:00	Joliet	IL	Other
80155	8/1/17 06:15	Columbus (North)	GA	Fireball
80156	8/1/17 02:45	Corcoran	MN	Light
80157	8/1/17 02:00	Moreno Valley	CA	Other
80158	8/1/17 01:00	Bradenton	FL	Other

80159 rows × 4 columns

6 – Manter os registros de Shapes mais populares (com mais de 1000 ocorrências).

```
ocorrencias_shape = '''
  SELECT Shape, COUNT(Shape) as Ocorrencia
  FROM excluir_col
  GROUP BY Shape
  HAVING Ocorrencia > 1000
'''

filtro_registro = pandasql.sqldf(ocorrencias_shape)
```

Aqui selecionamos o Shape e o COUNT (Shape) do dataframe excluir_col, agrupamos o Shape que tenham o número de ocorrências maior que 1000.

Recebemos como resultado uma pesquisa mais filtrada:

	Shape	Ocorrencia
0	Changing	2275
1	Chevron	1041
2	Cigar	1896
3	Circle	9313
4	Cylinder	1367
5	Diamond	1313
6	Disk	4299
7	Fireball	7535
8	Flash	1684
9	Formation	2863
10	Light	18877
11	Other	5931
12	Oval	3860
13	Rectangle	1436
14	Sphere	5882
15	Triangle	8330

7 – Retornar o dataframe criado com os campos Date/ Time, City, State e Shape comparando com a coluna Shape criada anteriormente.

```
filtro_final = '''
  SELECT excluir_col.*
  FROM excluir_col, filtro_registro
  WHERE filtro_registro.Shape = excluir_col.Shape
'''

df_final = pandasql.sqldf(filtro_final)
```

Agora no WHERE comparamos as colunas Shape de filtro_registro que foi criada no tópico anterior e a coluna Shape de excluir_col, o exemplo que retiramos 3 campos com o DROP.

E nosso dataframe final terá o seguinte resultado:

	Date / Time	City	State	Shape
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
...	...	...	...	...
77897	8/1/17 14:00	Joliet	IL	Other
77898	8/1/17 06:15	Columbus (North)	GA	Fireball
77899	8/1/17 02:45	Corcoran	MN	Light
77900	8/1/17 02:00	Moreno Valley	CA	Other
77901	8/1/17 01:00	Bradenton	FL	Other

77902 rows × 4 columns

8 – Crie um novo DataFrame

```
df_final.to_csv('df_OVNI_limpo.csv')
```

Referências

Pandas sql - https://pypi.org/project/pandasql/

Pandas - https://pandas.pydata.org/