Relatório UFO Data – Parte 2

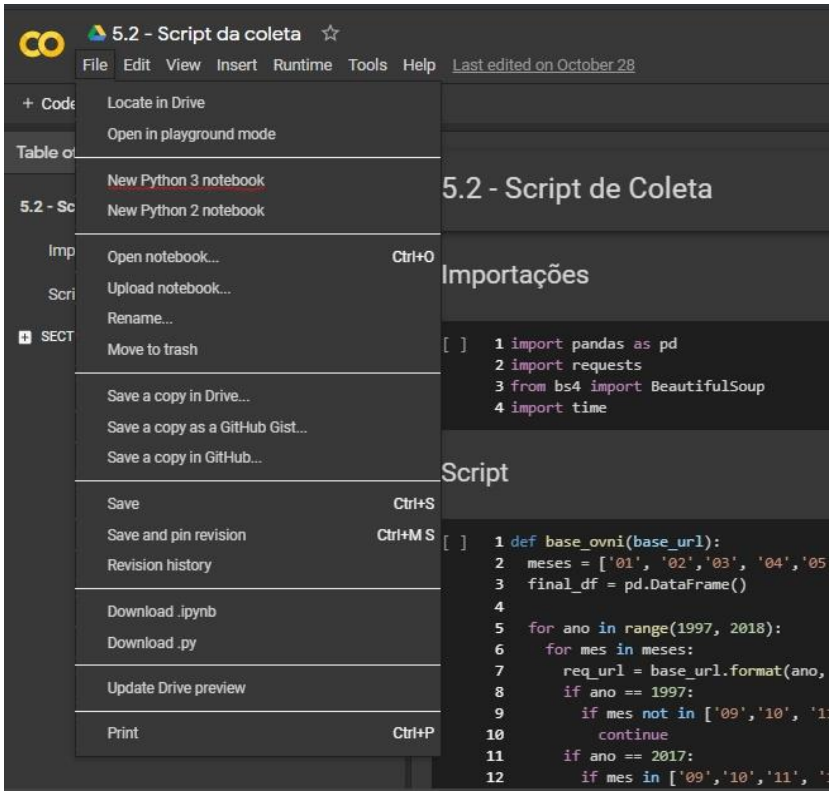
Levi Alves de Freitas Junior

Problema

Criar uma análise utilizando banco de dados, utilizando a base de dados requisitada no primeiro relatório, e utilizar o pandasql para a exploração dos dados. Vamos utilizar a base gerada no primeiro relatório e explora-la.

Para a primeira parte vamos criar um novo notebook no colab research. Siga as instruções a seguir e veja as imagens ilustrativas:

1 – Logo após terminar as instruções da primeira etapa de análise dos relatos dos ovni's, crie um novo notebook no colab, acessando no menu a opção File e logo após vá em New Python 3 notebook.



2 – 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 para instalar o pandassql no colab, logo após inicialize a célula no colab para importar o pandas e o pandas sql,

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

3 – Agora vamos importar a base de dados

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

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

#df.columns = ['Date/Time','City','State','Shape','Duration','Summary','Posted']

linhas = df['Date / Time'].size

print("Linhas: ",linhas)
```

Deletamos com o comando ‘del’ a coluna não nomeada pois não será necessária, então vamos imprimir a quantidade de linhas do nosso dataframe.

O resultado deve ser 99705 linhas.

4 – Vamos organizar os dados e remover campos em branco

```
estados = """
   SELECT State, COUNT(State) as qtd_rel FROM df WHERE State != ' ' GROUP BY State ORDER BY COUNT(State) desc ;
 """
result_consulta = pandasql.sqldf(estados)
result_consulta
```

Fazemos uma pesquisa utilizando o pandasql, os estados com mais relatos de aparições de ovni’s e organizamos em ordem decrescente.

O estado com mais relato é CA – Califórnia com 11403 relatos.

5 – Neste momento vamos analisar somente os casos relatados nos Estados Unidos, com o seguinte código vamos comparar com a base de dados para filtrar e retornar os relatos.

```
#Cria um array com os estados da coluna estado.

lista_usa = ['AK','AL','AR','AZ','CA','CO','CT','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']

lista = pd.DataFrame(lista_usa, columns=['Estados_Unidos'])
```

Criamos uma lista com as iniciais do estado e armazenamos esta lista à variável lista_usa e logo após criamos uma coluna de um dataframe e armazenamos na variável lista.

Com a coluna 'Estados_Unidos' do dataframe criada podemos agora comparar o dataframe original – 'df' com a coluna gerada e retorna somente os dados dos estados que constam na lista_usa. Então fazemos a seguinte query.

```
consulta_est_validos = '''
  SELECT df.* FROM df, lista
  WHERE State = Estados_Unidos
'''

limit_usa = pandasql.sqldf(consulta_est_validos)
```

6 – Consulta e filtragem das cidades com 10 relatos ou mais

```
cidades_agrup = """
    SELECT State, Shape, City, COUNT(*) as total_posts FROM limit_usa GROUP BY City;
  """
result_cidades_agrup = pandasql.sqldf(cidades_agrup)

nova_query_usa = """
    SELECT State, Shape, City, total_posts
    FROM result_cidades_agrup
    WHERE City != 'unknown' AND total_posts >= 10
    ORDER BY total_posts desc;
    """
sql_cidades_usa = pandasql.sqldf(nova_query_usa)
```

Fazemos a primeira query para agrupar as cidades e criamos uma subquery para tirar os campos não nomeados and que o total de posts (relatos) seja igual ou maior que 10.

O resultado desta query deve ser como no modelo abaixo.

	State	Shape	City	total_posts
0	AZ	Flash	Phoenix	558
1	WA	Changing	Seattle	548
2	OR	Circle	Portland	480
3	NV	Rectangle	Las Vegas	473
4	CA	Rectangle	San Diego	394
...	...	...	...	...
1833	AZ	Changing	Winslow	10
1834	MT	Flash	Wolf Point	10
1835	TX	Circle	Woodville	10
1836	MA	Teardrop	Yarmouth	10
1837	CA	Disk	Yucca Valley	10

1838 rows × 4 columns

7 – Porque esta é a cidade que mais tem relatos.

Phoenix - No dia 13 de março de 1997, o céu do estado do Arizona e de Nevada, encheram de luzes, milhares de pessoas presenciaram o ocorrido, as pessoas afirmaram ver algo em formato triangular vagando pelo céu na cidade de Phoenix, capital do Arizona e cidade com mais relatos na análise de todos os casos de aparecimento de OVNI's, o próprio governador presenciou e afirmou que seria um "objeto de outro mundo". Milhares de pessoas registraram essas aparições isso explica ser um dos locais com mais relatos.

8 – Busca nos estados com maior número de relato, juntamente buscando as cidades que tenham um número de relatos superior a 10 relatórios.

```
estados_usa_mais_rel = """
    SELECT State, MAX(qtd_rel) as qtd_rel FROM result_consulta;
    """
sql_estados = pandasql.sqldf(estados_usa_mais_rel)

cid_mais_rel = """
  SELECT sql_cidades_usa.City, sql_cidades_usa.total_posts, sql_cidades_usa.Shape
  FROM sql_estados, sql_cidades_usa
  WHERE sql_cidades_usa.State = sql_estados.State AND sql_cidades_usa.total_posts > 10 AND sql_cidades_usa.Shape NOT IN (' ', 'Unknown')
"""

result_final_cidades = pandasql.sqldf(cid_mais_rel)
```

O resultado deve ser o seguinte rodando result_final_cidades.

	City	total_posts	Shape
0	San Diego	394	Rectangle
1	Los Angeles	379	Changing
2	Sacramento	242	Changing
3	San Jose	223	Other
4	San Francisco	200	Chevron
...	...	...	...
221	Imperial Beach	11	Fireball
222	Miramar	11	Formation
223	Nipomo	11	Fireball
224	Pismo Beach	11	Disk
225	West Los Angeles	11	Triangle

226 rows × 3 columns

O projeto encontra-se no GitHub – para eventual consulta.

https://github.com/LeviAFJunior/Analise_Dados_Base_ovni/tree/master/Base_Ovni_parte2