Relatório (Limpeza de Dados)

O problema nesta parte do projeto era fazer uma limpeza nos dados, tratando os dados faltosos e eliminando dados desnecessários para a análise. Definindo e categorizando dados uteis para o projeto. Todo o projeto foi realizado no plataforma colab do GOOGLE.

1ª PARTE:

Como start para o projeto foi importado as bibliotecas necessária para a realização da limpeza dos dados. Neste caso foi somente necessário importar a biblioteca pandas para fazer a limpeza dos dados.

```
[ ]  1 # Importação das bibliotecas necessarias
     2 import pandas as pd
```

2ª PARTE:

Em seguida foi necessário carregar o DataFrame original de OVINS na extensão CSV onde contém os seus relatos, ‘Estados’ onde foi supostamente visto um OVIN e etc... . E em seguida na *linha 4* verificamos se o arquivo CSV foi carregado .

```
1 #1 Carregando dados CSV
2 df_ovni = pd.read_csv('https://raw.githubusercontent.com/oliveirafhm/data_science/master/OVNIS.csv')
3 df_ovni
```

Vamos verificar ?

	Date / Time	City	State	Shape	Duration	Summary	Posted
0	1/29/97 23:15	East Greenwich	RI	Disk	5 minutes	I witnessed a UFO which may be of the Lazar"Sp...	3/7/98
1	1/26/97 22:00	Flagstaff	AZ	Light	six minutes	It traveled at about the same speed we were go...	3/19/09
2	1/25/97 21:00	Marion	WI	Triangle	2 minutes	On a camping trip 3 triangle shaped objects we...	3/7/98
3	1/25/97 06:00	Mount Hope/Binbrook (Canada)	ON	Disk	1/2 hour	A large disk or saucer type object, approximat...	4/28/01
4	1/24/97 19:00	Alta	UT	Other	3 hours plus	A "moving star" similar to a satellite stopped...	3/7/98
...	...	...	...	...	...	...	...
102386	12/1/17 17:00	Foyil	OK	Formation	All night	At dusk my wife and I noticed a star that look...	12/10/17
102387	12/1/17 04:00	Chesapeake	VA	Light	5 minutes	Light was moving at a constant speed, vanished...	12/4/17
102388	12/1/17 04:00	Boise	ID	Cigar	10 minutes	Large cigar-shaped UFO with visible cabin lights.	12/4/17
102389	12/1/17 02:06	Ras Al khaimah (Oman/UAE)	NaN	Light	10 seconds	UFO PASSING.	12/4/17
102390	12/1/17 01:00	Wasilla	AK	Flash	25 minutes	Flashing, fast moving light over Alaska.	12/4/17

102391 rows × 7 columns

SUPINPA deu certo.

3ª PARTE:

Ótimo! Agora que o aquivo (CSV) está carregado, nosso primeiro passo é eliminar os dados faltosos. Para isso selecionaremos as colunas que queremos eliminar os dados *missing* e aplicamos o comano (*.notna()*).Onde esta função verifica se o objeto selecionado contém valores nulo ou ausentes.

No nosso caso selecionamos três colunas 'City', 'State' e 'Shape'. Para eliminar os dados ausentes ou nulos.

```
df_ovni = [df_ovni['City'].notna()]
df_ovni = [df_ovni['State'].notna()]
df_ovni = [df_ovni['Shape'].notna()]
```

CURIOSIDADE: O comando "dropna" elimina todos os dados faltosos do DataFrame.

```
df_ovni = df_ovini.dropna()
```

4ª PARTE:

Em seguida precisamos manter no nosso DataFrame somente os estados que contém nos Estados Unidos - USA. Para isso, primeiro precisamos carregar um DataFrame com todos os Estados dos USA. Aqui carregaremos um arquivo (CSV) com os dados que precisamos.

OBS:. (Esta é apenas uma Opção, se preferir também pode criar uma lista vazia com todos os estados dos Estados Unidos).

```
df_states_usa = pd.read_csv('https://raw.githubusercontent.com/oliveirafhm/data_science/master/usa_states.csv')
df_ovni['State']
```

Vamos testar se está funcionando, carregar os dados ?

	State	Abbreviation
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA
5	Colorado	CO
6	Connecticut	CT
7	Delaware	DE
8	District of Columbia	DC
9	Florida	FL
10	Georgia	GA

OPA, os dados foram carregados com sucesso!. Estamos progredindo bem. Agora vamos comparar a coluna ‘Abbreviation’ (que contém a abreviatura de todos os estados de USA) com ‘State’ para selecionar e filtrar somente os estados que residem em USA.

```
filtro2 = []
for reg in list(df_ovni['State']):
  if reg in list(df_states_usa['Abbreviation']):
    filtro2.append(True)
  else:
    filtro2.append(False)
df_ovni = df_ovni[filtro2]
# Amostra filtrada somente com estados dos 'Estados Unidos'
df_ovni
```

Inicialmente criamos uma lista vazia para armazenar a comparação dos dados que precisamos. Em seguida fazemos uma comparação entre as colunas selecionadas ‘State’ & ‘Abbreviation’ para ver se os dados de uma coluna batem corretamente com a outra.

Vamos ver os resultados esperados e foram o que esperávamos?

```
# Amostra filtrada somente com estados dos 'Estados Unidos'
df_ovni
```

ANTES : & DEPOIS :

State
RI
AZ
WI
ON
UT
...
OK
VA
ID
NaN
AK

State
RI
AZ
WI
UT
RI
...
NY
OK
VA
ID
AK

Comparando os dados, foram eliminados os dados faltosos dos três campos selecionados. Olha que Blz. Para fins de comparação se somente os dados de ‘State’ & ‘Abbreviation’ tão corretos, vejamos uma comparação dos dados ANTES e DEPOIS do tamanho dos dados.

ANTES & DEPOIS

```
(102391, 7)          (90718, 7)
```

5ª PARTE:

Para o próximo passo vamos deletar os dados que não tem relevância e nem sentido para nossa análise. Acessando a função *.drop* do Pandas e selecionando as colunas desejadas [‘Duration’, ‘Summary’, ‘Posted’] no eixo = 1 referente a colunas deletadas.

```
27 df = df_ovni.drop(['Duration','Summary','Posted'],axis=1)
```

Vamos rodar nosso DataFrame pra ver qual o resultado que saiu?

```
28 df
```

	Date / Time	City	State	Shape
0	1/29/97 23:15	East Greenwich	RI	Disk
1	1/26/97 22:00	Flagstaff	AZ	Light
2	1/25/97 21:00	Marion	WI	Triangle
4	1/24/97 19:00	Alta	UT	Other
5	1/23/97 18:30	North Kingstown	RI	Triangle
...	...	...	...	...
102385	12/1/17 17:00	New Rochelle	NY	Sphere
102386	12/1/17 17:00	Foyil	OK	Formation
102387	12/1/17 04:00	Chesapeake	VA	Light
102388	12/1/17 04:00	Boise	ID	Cigar
102390	12/1/17 01:00	Wasilla	AK	Flash

86838 rows × 4 columns

Olha que BLZ! As colunas que deletamos sumiram do nosso DataFrame. E se observar bem em baixo do DataFrame mostra que temos agora 86838 dados e somente 4 colunas com os uteis para nossa análise.

6ª PARTE:

E para finalizar vamos mostras somente os registros de Shapes mais populares > 1000 ocorrências. Primeiro verificamos a frequência dos Shapes que o objeto resultante estará em ordem decrescente para que o primeiro elemento seja o elemento que ocorre com mais frequência.

```
2 shape = df['Shape'].value_counts()
```

Em seguida usaremos a função *.value_counts* para Redefinir o índice do DataFrame e usa-ló de forma padrão.

```
5 shape = shape.reset_index()
```

Em seguida definimos uma condição para mostras somente os Shapes que contém os relatos populares

```
8 shape_popular =  shape[shape['Shape']> 1000]['index']
```

Veremos o resultado. Será que deu certo?

0	Light
1	Circle
2	Triangle
3	Fireball
4	Unknown
5	Other
6	Sphere
7	Disk
8	Oval
9	Formation
10	Changing
11	Cigar
12	Flash
13	Rectangle
14	Cylinder
15	Diamond
16	Chevron

Calma pequeno gafanhoto, deu tudo certo!!!

7ª PARTE:

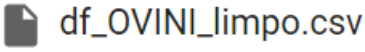
Para finalizarmos esse projeto. Agora vamos criar um arquivo (CSV).

```
1 # Salvar o 'Df' com os dados limpos
2 shape_popular.to_csv('df_OVINI_limpo.csv')
```

Vamos visualizar o DataFrame CSV?

SIM!!

Esse é o espirito garoto.



Veja que o nome que foi definido (df_OVNi_limpo.csv), foi criado!

Finalzad...

Espera um pouco, como ficou o DataFrame final ?

Boa pergunta, vejamos como ficou o DataFrame final.

...

Pronto! Ai está o DataFrame final com a limpeza realizada

0	Light
1	Circle
2	Triangle
3	Fireball
4	Unknown
5	Other
6	Sphere
7	Disk
8	Oval
9	Formation
10	Changing

Show 10 ▾ per page

0	Light
11	Cigar
12	Flash
13	Rectangle
14	Cylinder
15	Diamond
16	Chevron

Show 10 ▾ per page

LINK do projeto: https://github.com/samuelflopes/Coleta.git

FIM