

5.8 - Acréscimo de variáveis

A tarefa desse módulo consiste em acrescentar variáveis e organizá-las para que possamos obter dados mais filtrados.

1º passo: Importar as bibliotecas:

```
[12] 1 !pip install -U pandasql
```

```
[79] 1 import pandas as pd  
    2 import pandasql
```

2º passo: Carregar o arquivo *df_OVNI_limpo.csv* (arquivo gerado após a limpeza de dados efetuada na atividade 5.7) em um dataframe e deletar a coluna *'Unnamed: 0'* que restou da atividade passada.

```
1 df = pd.read_csv('df_OVNI_limpo.csv')  
2 del df['Unnamed: 0']
```

3º passo: Dividir o conteúdo da coluna *Date / Time* em duas novas colunas no mesmo dataframe e deletar a coluna *Date / Time*.

Adicionamos a coluna *'Date / Time'* do nosso dataframe à variável chamada **df_dt**. Além disso, utilizamos a função **pd.to_datetime** para converter os dados para o formato de data.

```
1 df_dt = pd.to_datetime(df['Date / Time'])
```

Em seguida, utilizamos a função **strftime** para pegar os dados separados. Com o **%H** pegamos a hora, com o **%M** pegamos os minutos e **%d**, **%m** e **%y** são dia mês e ano, respectivamente.

Logo em seguida, adicionamos ao dataframe.

```
3 hr = df_dt.dt.strftime('%H:%M')  
4 dt = df_dt.dt.strftime('%d/%m/%y')  
5 df['Sight_Date'] = dt  
6 df['Sight_Time'] = hr
```

4º passo: Deletar a coluna *'Date / Time'*

```
[71] 1 del df['Date / Time']
```

5º Passo: Criar uma nova coluna chamada 'Sight_Weekday' para colocar os dias da semana.

```
[73] 1 dias_semana = pd.to_datetime(df['Sight_Date']).dt.weekday
      2 df['Sight_Weekday'] = dias_semana
```

Utilizamos a função **to_datetime** novamente para representar os dias em números de 0 a 6. Cada número representa um dia da semana, iniciando em segunda-feira (representado pelo numeral 0) e finalizando no domingo (representado pelo numeral 6).

| | City | State | Shape | Sight_Date | Sight_Time | Sight_Weekday |
|------------------------|-----------------|-------|-----------|------------|------------|---------------|
| 0 | East Greenwich | RI | Disk | 29/01/97 | 23:15 | 2 |
| 1 | Flagstaff | AZ | Light | 26/01/97 | 22:00 | 6 |
| 2 | Marion | WI | Triangle | 25/01/97 | 21:00 | 5 |
| 3 | Alta | UT | Other | 24/01/97 | 19:00 | 4 |
| 4 | North Kingstown | RI | Triangle | 23/01/97 | 18:30 | 3 |
| ... | ... | ... | ... | ... | ... | ... |
| 80024 | New Rochelle | NY | Sphere | 01/12/17 | 17:00 | 3 |
| 80025 | Foyil | OK | Formation | 01/12/17 | 17:00 | 3 |
| 80026 | Chesapeake | VA | Light | 01/12/17 | 04:00 | 3 |
| 80027 | Boise | ID | Cigar | 01/12/17 | 04:00 | 3 |
| 80028 | Wasilla | AK | Flash | 01/12/17 | 01:00 | 3 |
| 80029 rows × 6 columns | | | | | | |

6º Passo: Criar uma lista com os dias da semana

```
[89] 1 dias = {  
2     0: 'Segunda-Feira',  
3     1: 'Terça-Feira',  
4     2: 'Quarta-Feira',  
5     3: 'Quinta-Feira',  
6     4: 'Sexta-Feira',  
7     5: 'Sábado',  
8     6: 'Domingo'  
9 }
```

7º Passo: Substituir os números da coluna 'Sight_Weekday' pelos dias da lista que acabamos de criar

Para isso, criamos um for para percorrer a lista substituindo-os.

```
[90] 1 dias_col = [dias[i] for i in list(df['Sight_Weekday'])]  
2 df['Sight_Weekday'] = dias_col
```

8º Passo: Adicionar duas novas colunas chamadas 'Sight_Day' e 'Sight_Month' para indicar o dia e mês.

```
[91] 1 dia = df_dt.dt.strftime('%d')  
2 mes = df_dt.dt.strftime('%m')  
3 df['Sight_Day'] = dia  
4 df['Sight_Month'] = mes
```

9º e Último Passo: salvar o dataframe resultante em um novo arquivo CSV

```
[92] 1 df.to_csv('df_OVNI_preparado.csv')
```