

Problema

Organizar as colunas do DataFrame dividir as informações de data e horário para colunas diferentes e retornar os dias da semana de acordo com a data.

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

1 – Após criar o notebook, podemos importar as bibliotecas para realizarmos os procedimentos.

```
import pandas as pd

df = pd.read_csv('df_OVNI_limpo.csv')

#Retirando coluna indesejada
del df['Unnamed: 0']
```

Importamos a biblioteca pandas que usamos ao longo dos tutoriais, em seguida carregamos nosso dataframe df_OVNI_limpo que criamos no tópico anterior, e por fim retiramos no bloco a coluna do df que não nos interessa.

2 – Vamos criar as colunas para dividir a data e o horário, inserimos a coluna Sight_Date para data e a coluna Sight_Time para o horário. Veja o exemplo a seguir:

```
#Criando coluna no dataframe somente ano
df['Sight_Date'] = pd.to_datetime(df['Date / Time']).dt.strftime('%m/%d/%y')
#Criando coluna no dataframe somente hora
df['Sight_Time'] = pd.to_datetime(df['Date / Time']).dt.strftime('%H:%M')
```

No exemplo criamos as colunas de data e hora no final do dataframe e como resultado receberemos:

	Date / Time	City	State	Shape	Sight_Date	Sight_Time
0	9/30/97 22:00	Madison	WI	Light	09/30/97	22:00
1	9/28/97 23:15	San Francisco	CA	Triangle	09/28/97	23:15
2	9/27/97 23:00	Egan	SD	Other	09/27/97	23:00
3	9/27/97 05:00	Crestwood	KY	Disk	09/27/97	05:00
4	9/25/97 22:00	Clearfield	UT	Triangle	09/25/97	22:00
...
77897	8/1/17 14:00	Joliet	IL	Other	08/01/17	14:00
77898	8/1/17 06:15	Columbus (North)	GA	Fireball	08/01/17	06:15
77899	8/1/17 02:45	Corcoran	MN	Light	08/01/17	02:45
77900	8/1/17 02:00	Moreno Valley	CA	Other	08/01/17	02:00
77901	8/1/17 01:00	Bradenton	FL	Other	08/01/17	01:00

3 – Essa etapa somente iremos retirar a coluna de Date Time, pois não iremos mais utilizá-la.

```
df = df.drop(['Date / Time'], axis=1)
```

E como resultado a tabela a seguir:

	City	State	Shape	Sight_Date	Sight_Time
0	Madison	WI	Light	09/30/97	22:00
1	San Francisco	CA	Triangle	09/28/97	23:15
2	Egan	SD	Other	09/27/97	23:00
3	Crestwood	KY	Disk	09/27/97	05:00
4	Clearfield	UT	Triangle	09/25/97	22:00
...
77897	Joliet	IL	Other	08/01/17	14:00
77898	Columbus (North)	GA	Fireball	08/01/17	06:15
77899	Corcoran	MN	Light	08/01/17	02:45
77900	Moreno Valley	CA	Other	08/01/17	02:00
77901	Bradenton	FL	Other	08/01/17	01:00

4 – Agora Criaremos a coluna chamada Sight_Weekday para incluirmos os dias da semana de acordo com o Sight_Date para isso utilizaremos a coluna de data e uma função do pandas que retorna um valor entre 0 e 6, cada um para um dia da semana.

```
df['Sight_Weekday'] = pd.to_datetime(df['Sight_Date']).dt.dayofweek

dias = {
    0: 'Segunda-Feira',
    1: 'Terça-Feira',
    2: 'Quarta-Feira',
    3: 'Quinta-Feira',
    4: 'Sexta-Feira',
    5: 'Sábado',
    6: 'Domingo'
}

df = df.replace({'Sight_Weekday': dias})
```

Inicialmente criamos a coluna Sight_Weekday e com to_datetime transformamos os valores da coluna do dataframe de objeto para valores de data e hora e utilizamos a função dayofweek para retornar um valor dos dias da semana, porém, utilizamos uma lista para referenciar cada valor para um dia específico, começando na Segunda-Feira que a função dayofweek retornará 0, e com isso utilizamos o replace para tirarmos o valor numérico e inserimos os valores por extenso, de acordo com o dicionário criado e nomeado dias.

5 – Criar uma coluna somente para o dia e uma coluna somente para o mês.

Agora vamos adicionar o dia da semana para a coluna Sight_day, para isso utilizaremos as mesmas funções do tópico 2.

```
#Coluna para criar o dia da semana
df['Sight_Day'] = pd.to_datetime(df['Sight_Date']).dt.strftime('%d')
#Coluna para criar o mes
df['Sight_Month'] = pd.to_datetime(df['Sight_Date']).dt.strftime('%m')
```

Foi alterado somente o parâmetro dentro de strftime, no primeiro, será retornado o dia com %d e no segundo com %m para inserirmos somente o mês. Passamos para cada coluna o dado específico da coluna Sight_date.

Finalizamos a inserção das colunas, agora vamos exportar nosso df para um arquivo externo, vamos nomeá-lo como df_OVNI_preparado.

```
df.to_csv('df_OVNI_preparado', index = False)
```

Referências

pandas - <https://pandas.pydata.org/pandas-docs/stable/>

strftime - <https://www.geeksforgeeks.org/python-pandas-series-dt-strftime/>

dayofweek-

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DatetimeIndex.dayofweek.html>