Data Cleaning Project

Robson Silva

Info dos dados:

Case Number	Location	Fatal (Y/N)	href
Date	Activity	Time	Case Number.1
Year	Name	Species	Case Number.2
Type	Sex	Investigator or Source	Original Order
Country	Age	pdf	Unnamed: 22
Area	Injury	Href formula	Unnamed: 23

Iniciando com o básico:

```
In [5]:   1  attacks.drop_duplicates(inplace=True)

In [6]:   1  attacks.reset_index(inplace=True)

In [8]:   1  attacks.drop(columns='index', inplace=True)

In [7]:   1  attacks.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6312 entries, 0 to 6311
Data columns (total 25 columns):
index               6312 non-null int64
Case Number         6310 non-null object
Date                6302 non-null object
Year                6300 non-null float64
Type                6298 non-null object
Country             6252 non-null object
Area                5847 non-null object
Location            5762 non-null object
Activity            5758 non-null object
Name                6092 non-null object
Sex                 5737 non-null object
Age                 3471 non-null object
Injury              6274 non-null object
Fatal (Y/N)         5763 non-null object
```

+bike

Colunas deletáveis:

```
href formula              6301 non-null object
href                      6302 non-null object
Case Number.1             6302 non-null object
Case Number.2             6302 non-null object
original order            6309 non-null float64
Unnamed: 22               1 non-null object
Unnamed: 23               2 non-null object
dtypes: float64(2), int64(1), object(22)
memory usage: 1.2+ MB
```

```
n [9]:  1  attacks.drop(columns=['Unnamed: 22', 'Unnamed: 23'], axis=1, inplace=True)
```

Ampliando a visualização.

```
In [10]:   1  pd.options.display.max_rows
Out[10]:  60

In [159]:  1  pd.options.display.max_rows = 200
```

Organizando o trabalho.

Procurei por colunas mais organizadas e decidi voltar no final do projeto.
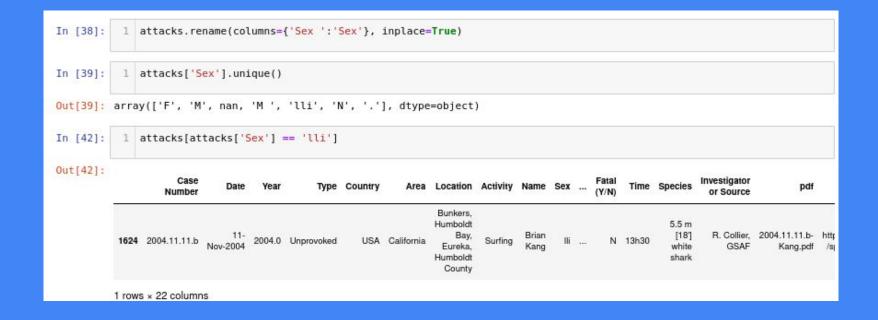
- Country
- Area
- Location

Substituição manual de alguns valores da Coluna Fatal (Y/N).

```python
attacks['Fatal (Y/N)'].replace({'M':'N',' N':'N', 'N ':'N', 'y':'Y','2017':np.nan}, inplace=True)
```

Coluna 'Activity'

```
In [26]:   1   attacks['Activity'].describe()

Out[26]:   count          5758
           unique         1532
           top         Surfing
           freq            971
           Name: Activity, dtype: object

In [31]:   1   attacks['Activity'].unique()

Out[31]:   array(['Paddling', 'Standing', 'Surfing', ...,
                 'Crew swimming alongside their anchored ship',
                 '4 men were bathing', 'Wreck of  large double sailing canoe'],
                 dtype=object)

In [30]:   1   attacks['Activity'].isnull().sum()

Out[30]:   554
```

Coluna 'Sex'

```
In [38]:  1  attacks.rename(columns={'Sex ':'Sex'}, inplace=True)
```

```
In [39]:  1  attacks['Sex'].unique()
```

```
Out[39]:  array(['F', 'M', nan, 'M ', 'lli', 'N', '.'], dtype=object)
```

```
In [42]:  1  attacks[attacks['Sex'] == 'lli']
```

Out[42]:

	Case Number	Date	Year	Type	Country	Area	Location	Activity	Name	Sex	...	Fatal (Y/N)	Time	Species	Investigator or Source	pdf	
1624	2004.11.11.b	11-Nov-2004	2004.0	Unprovoked	USA	California	Bunkers, Humboldt Bay, Eureka, Humboldt County	Surfing	Brian Kang	lli	...	N	13h30	5.5 m [18] white shark	R. Collier, GSAF	2004.11.11.b-Kang.pdf	http /s

1 rows × 22 columns

attacks['Sex'].replace({'M ':'M', 'N':'M',}, inplace=True)
Mais de 500 linhas nulas.

'Href' e 'href formula' colunas.

```
In [49]:  1  href = attacks[attacks['href formula'] != attacks['href']]
```

```
In [50]:  1  href.reset_index()
          2
```

Out[50]:

	index	Case Number	Date	Year	Type	Country	Area	Location	Activity	Name	...	Fatal (Y/N)	Time	Species	Investigator or Source	
0	50	2018.01.13	13-Jan-2018	2018.0	Unprovoked	AUSTRALIA	New South Wales	Martin Islet	Free diving	Callum Stewart	...	N	NaN	White shark, 3.5 m	B. Myatt, GSAF	20 S

70 rows × 23 columns

Eliminando 'href'

```python
for i in range(len(href)):
    print(i)
    print(href['href'].iloc[i])
    print(href['href formula'].iloc[i])
```

```
0
http://sharkattackfile.net/spreadsheets/pdf_directory/http://sharkattackfile.net/spreadsheets/pdf_directory/2018.0
1.13-Stewart.pdf
http://sharkattackfile.net/spreadsheets/pdf_directory/2018.01.13-Stewart.pdf
1
http://sharkattackfile.net/spreadsheets/pdf_directory/http://sharkattackfile.net/spreadsheets/pdf_directory/2017.0
8.27-Brundler.pdf
http://sharkattackfile.net/spreadsheets/pdf_directory/2017.08.27-Brundler.pdf
2
http://sharkattackfile.net/spreadsheets/pdf_directory/http://sharkattackfile.net/spreadsheets/pdf_directory/2017.0
6.05-FrenchPolynesia.pdf
http://sharkattackfile.net/spreadsheets/pdf_directory/2017.06.05-FrenchPolynesia.pdf
3
```

```python
In [ ]:    attacks.drop(columns='href', axis=1, inplace=True)
```

```python
In [56]:   attacks.rename(columns={'href formula':'Link'}, inplace=True)
```

```python
In [57]:   attacks.dropna(thresh=9, axis=0, inplace=True)
```

```python
In [58]:   attacks.reset_index(inplace=True)
```

Trabalhando com 'Type'.

```
In [63]:   1  attacks['Type'].replace({'Boatomg':'Boating'}, inplace=True)

In [65]:   1  attacks['Type'].unique()

Out[65]: array(['Boating', 'Unprovoked', 'Invalid', 'Provoked', 'Questionable',
                'Sea Disaster', nan, 'Boat'], dtype=object)
```

Type possui apenas valores 4 nulos.

Colunas: Case Numbers

```
In [76]:    1   attacks[attacks['Case Number.1'] != attacks['Case Number']]
```

Out[76]:

	index	Case Number	Date	Year	Type	Country	Area	Location	Activity	Name	...	Injury	Fatal (Y/N)	Time	Species
34	34	2018.04.03	03-Apr-2018	2018.0	Unprovoked	SOUTH AFRICA	Eastern Cape Province	St. Francis Bay	Surfing	Ross Spowart	...	Lacerations to left knee & lower leg	N	15h00	White shark Tra

24 rows × 22 columns

```
In [78]:    1   attacks['Case Number'].iloc[5488] = '1905.09.06'

In [79]:    1   attacks[attacks['Case Number'].isnull()]
```

Out[79]:

index	Case Number	Date	Year	Type	Country	Area	Location	Activity	Name	...	Injury

0 rows × 22 columns

```
In [80]:    1   attacks.drop(columns=['Case Number.1','Case Number.2'], inplace=True)
```

Coluna 'Year'

```
In [95]:  1  attacks['Year'] = attacks['Year'].apply(int)
```

```
In [87]:  1  attacks[attacks['Year'].isnull()]
```

Out[87]:

	index	Case Number	Date	Year	Type	Country	Area
187	187	2017.01.08.R	Reported 08-Jan-2017	NaN	Invalid	AUSTRALIA	Queensland
6079	6079	1836.08.19.R	Reported 19-Aug-1836	NaN	Unprovoked	ENGLAND	Cumberland

```
In [90]:  1  attacks['Year'].iloc[187] = 2017
```

```
In [93]:  1  attacks['Year'].iloc[6079] = 1836
```

```
[94]:  1  attacks[attacks['Year'] == 0]
```

[94]:

	index	Case Number	Date	Year	Type	Country	Area
6177	6177	0000.0214	Ca. 214 B.C.	0.0	Unprovoked	NaN	Ionian Sea
6178	6178	0000.0336	Ca. 336.B.C.	0.0	Unprovoked	GREECE	Piraeus

6297	6.0	ND.0005	Before 1903	0	Unp
6298	5.0	ND.0004	Before 1903	0	Unp

Limpando 'Year'

```
In [371]:  1  def rob_string(x):
           2      if re.findall(r'.*B\.C*.', x):
           3          return int(''.join(re.findall(f'\d+', x)))*-1
           4      elif re.findall(r'.*[bB]efore.*', x):
           5          return ''.join(re.findall(f'\d+', x))
           6      else:
           7          return 0
```

```
In [368]:  1  texto = 'Ca. 214 B.C.'
```

```
In [369]:  1  rob_string(texto)
```

```
Out[369]:  -214
```

```
In [370]:  1  tex = 'Before Mar-1956'
```

```
In [362]:  1  rob_string(tex)
```

```
Out[362]:  1956
```

```
In [372]:  1  attacks['Teste'] = attacks['Date'].apply(rob_string)
```

```
In [373]:  1  for i in range(len(attacks['Year'])):
           2      if attacks['Year'][i] == 0:
           3          attacks['Year'][i] = attacks['Teste'][i]
```

```
           4
           5  attacks.loc[ attacks['Year']==0, 'Year' ] = 'Teste'
```

Coluna Year, mais limpa, sem nenhum elemento nulo.

```
In [459]:   1   attacks['Year'].unique()

Out[459]: array([2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008,
                  2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000, 1999, 1998, 1997,
                  1996, 1995, 1984, 1994, 1993, 1992, 1991, 1990, 1989, 1969, 1988,
                  1987, 1986, 1985, 1983, 1982, 1981, 1980, 1979, 1978, 1977, 1976,
                  1975, 1974, 1973, 1972, 1971, 1970, 1968, 1967, 1966, 1965, 1964,
                  1963, 1962, 1961, 1960, 1959, 1958, 1957, 1956, 1955, 1954, 1953,
                  1952, 1951, 1950, 1949, 1948, 1848, 1947, 1946, 1945, 1944, 1943,
                  1942, 1941, 1940, 1939, 1938, 1937, 1936, 1935, 1934, 1933, 1932,
                  1931, 1930, 1929, 1928, 1927, 1926, 1925, 1924, 1923, 1922, 1921,
                  1920, 1919, 1918, 1917, 1916, 1915, 1914, 1913, 1912, 1911, 1910,
                  1909, 1908, 1907, 1906, 1905, 1904, 1903, 1902, 1901, 1900, 1899,
                  1898, 1897, 1896, 1895, 1894, 1893, 1892, 1891, 1890, 1889, 1888,
                  1887, 1886, 1885, 1884, 1883, 1882, 1881, 1880, 1879, 1878, 1877,
                  1876, 1875, 1874, 1873, 1872, 1871, 1870, 1869, 1868, 1867, 1866,
                  1865, 1864, 1863, 1862, 1861, 1860, 1859, 1858, 1857, 1856, 1855,
                  1853, 1852, 1851, 1850, 1849, 1847, 1846, 1845, 1844, 1842, 1841,
                  1840, 1839, 1837, 1836, 1835, 1834, 1832, 1831, 1830, 1829, 1828,
                  1827, 1826, 1825, 1823, 1822, 1819, 1818, 1817, 1816, 1815, 1812,
                  1811, 1810, 1808, 1807, 1805, 1804, 1803, 1802, 1801, 1800, 1797,
                  1792, 1791, 1788, 1787, 1786, 1785, 1784, 1783, 1780, 1779, 1776,
                  1771, 1767, 1764, 1758, 1753, 1751, 1749, 1755, 1748, 1742, 1738,
                  1733, 1723, 1721, 1703, 1700, 1642, 1638, 1637, 1617, 1595, 1580,
                  1555, 1554, 1543,  500,   77,    5, -214, -336, -493, -725,    0,
```

'0' São 37 valores que serão feitos manualmente.

Limpando 'Name'

```
In [472]:   1  len(attacks['Name'].apply(lambda x:re.findall(r'.*male.*',x)).sum())

Out[472]:  809
```

```
Name: Name, Length: 0502, dtype: bool]

In [494]:   1  attacks['Sex'].isnull().sum()

Out[494]:  559
```

```
In [496]:   1  attacks.loc[(attacks['Name']==r'.*^male.*') & (attacks['Sex'].isnull())]

Out[496]:
```

index	Case Number	Date	Year	Type	Country	Area	Location	Activity	Name	...	Age	Injury	Fatal (Y/N)	Time	Spe

0 rows × 21 columns

Conclusões e melhorias