Manipulação de Arquivos e Strings

Semana 13 | Uso do Regex

Questão Dirigida

Como gravar em um arquivo CSV os dados da pandemia de COVID-19 por país desta página da Wikipedia?

Desenvolvimento do Projeto

Para desenvolver este projeto, vamos utilizar técnicas de raspagem de dados na Web (Web Scraping), com a biblioteca `requests`, `BeautifulSoup` e expressões regulares.

In [1]:
```
# download
import requests

# parsing do HTML
from bs4 import BeautifulSoup

# expressões regulares
import re
```

A incidência de COVID-19 por país pode ser encontrada na página da Wikipedia.

In [2]:
```
# URL de análise
url_pagina = 'https://pt.wikipedia.org/wiki/Portal:COVID-19'
```

In [3]:
```
# Utiliza a biblioteca requests para fazer o download da pagina HTML
pagina_html = requests.get(url_pagina)

# O Código 200 indica que o download foi bem sucedido
pagina_html
```

Out[3]:
```
<Response [200]>
```

In [4]:
```
# O Beautiful Soup
soup = BeautifulSoup(pagina_html.content, 'html.parser')
#print(soup.prettify())
```

Inspeção da página

Inspecionando o código-fonte da página HTML no navegador (Chrome, Firefox etc) podemos ver que a tabela de interesse contém a classe CSS: `wikitable sortable mw-collapsible no-ref`. Vamos utilizar esta classe CSS para restringir a busca pelos elementos de interesse.

In [5]:
```
tb_covid_pais = soup.find(class_='wikitable sortable mw-collapsible no-ref')
#tb_covid_pais
```

Opcionalmente, pode-se fazer o seguinte.

In [6]:
```
tbs_covid_pais = soup.find_all(class_='wikitable sortable mw-collapsible no-r
# Lembre-se que o método find_all retorna uma lista. No caso deste exemplo,
# só existe um elemento na lista
tb_covid_pais = tbs_covid_pais[0]
# tb_covid_pais
```

In [7]:
```
# Retorna uma lista com todas as linhas da tabela
linhas_tb = tb_covid_pais.find_all('tr')
#linhas_tb
```

> **Atenção**: É sempre importante entender o tipo de uma variável.

In [8]:
```
type(linhas_tb)
```

Out[8]:
```
bs4.element.ResultSet
```

In [9]:
```
type(linhas_tb[0])
```

Out[9]:
```
bs4.element.Tag
```

Investigando os resultados

Primeiro país ocorre na posição 2.

In [10]:
```
linhas_tb[2].get_text()
```

Out[10]:
```
'\n Estados Unidos[a]\n33\xa0448\xa0422\n\n601\xa0079\n\n—\n\n[3]\n'
```

Último ocorre na posição 226.

In [11]:
```
linhas_tb[226].get_text()
```

Out[11]:
```
'\n Saba\n7\n\n0\n\n7\n\n[292]\n'
```

In [12]:
```
ini_tb = 2
fim_tb = 226
```

Brincando um pouco com esses objetos podemos ver que eles contém os dados que desejamos. No entanto, esses dados precisem ser tratados. Por exemplo: podemos ver que o resultado a seguir começa e termina com `\n` (caractere de nova linha); existe um espaço antes do nome do país; existem vários `\n` e `\xa0` (*line feed*) no meio da linha. Além disso, existem notas associadas ao nome do país e ao final da linha. Vamos cuidar de cada um desses problemas.

In [13]:
```
lin = linhas_tb[2].get_text()
lin
```

Out[13]:
```
'\n Estados Unidos[a]\n33\xa0448\xa0422\n\n601\xa0079\n\n—\n\n[3]\n'
```

Removendo os `\n` e os espaços no começo e ao final da linha.

O método `strip()` remove caracteres no começo e ao final da linha.

In [14]:
```
lin
```

Out[14]:
```
'\n Estados Unidos[a]\n33\xa0448\xa0422\n\n601\xa0079\n\n—\n\n[3]\n'
```

In [15]:
```
lin.strip("\n \t")
```

Out[15]:
```
'Estados Unidos[a]\n33\xa0448\xa0422\n\n601\xa0079\n\n—\n\n[3]'
```

Note que a variável `lin` não é alterada desta forma.

In [16]:
```
lin
```

Out[16]:
```
'\n Estados Unidos[a]\n33\xa0448\xa0422\n\n601\xa0079\n\n–\n\n[3]\n'
```

Para que a variável `lin` seja alterada, é necessário utilizar a atribuição.

In [17]:
```
lin = lin.strip("\n ")
lin
```

Out[17]:
```
'Estados Unidos[a]\n33\xa0448\xa0422\n\n601\xa0079\n\n–\n\n[3]'
```

Removendo os caracteres `\xa0` e `\n` no meio da string

Podemos utilizar o método `replace()` para substituir o caractere `\xa0` com a string vazia, para que o número de casos não esteja com espaços entre eles. Note que o caractere `\xa0` está em Unicode (que podem ser representados por hexadecimais), por isso necessitam do `u` antes do começo da string.

In [18]:
```
lin = lin.replace(u"\xa0", u"")
lin
```

Out[18]:
```
'Estados Unidos[a]\n33448422\n\n601079\n\n–\n\n[3]'
```

Como desejamos criar um arquivo CSV ao final do processo, vamos substituir um ou mais `\n` e `\t` que separa cada coluna da tabela por um `\t` .

In [19]:
```
lin = re.sub("[\n|\t]+","\t", lin)
lin
```

Out[19]:
```
'Estados Unidos[a]\t33448422\t601079\t–\t[3]'
```

Removendo as notas do nome do país e ao final da linha

Para remover as notas do nome do país, vamos utilizar a seguinte expressão regular: `r"\[[\d|a-z]+\]"` . A expressão pode ser lida da seguinte forma: selecione o texto de dígitos ou caracteres alfa-numéricos que estejam entre `[` e `]` .

In [20]:
```
lin
```

Out[20]:
```
'Estados Unidos[a]\t33448422\t601079\t–\t[3]'
```

In [21]:
```
lin = re.sub(r"\[[\d|a-z]+\]", "", lin)
lin
```

Out[21]:
```
'Estados Unidos\t33448422\t601079\t–\t'
```

Remove-se finalmente o `\t` ao final da string.

In [22]:
```
# Coluna Fontes
lin = lin.rstrip("\t")
lin
```

Out[22]:
```
'Estados Unidos\t33448422\t601079\t–'
```

In [23]:
```
lin = lin.replace("–", "0")
lin
```

Out[23]:
```
'Estados Unidos\t33448422\t601079\t0'
```

Criação de uma função de preparação da linha

Para se criar a função, vamos começar aplicando a sequência de passos para preparar a linha dos EUA e em seguida analisar se o que fizemos é suficiente para os outros casos.

In [24]:
```
def prep_linha_v1(linha):
    # Remove os caracteres \n no começo e final da string
    # Em seguida, remove os espaços
    linha = linha.strip("\n ")
    linha = linha.replace(u"\xa0", u"")
    linha = re.sub("[\n|\t]+","\t", linha)
    linha = re.sub(r"\[[\d|a-z]+\]", "", linha)
    linha = linha.strip("\t")
    linha = re.sub(r"–", "0", linha)
    return linha
```

Como esperado, a função funciona para os EUA.

In [25]:
```
eua = prep_linha_v1(linhas_tb[2].get_text())
eua
```

Out[25]:
```
'Estados Unidos\t33448422\t601079\t0'
```

Preste atenção nos tipos de dados da entrada das funções.

Chamada com String.

In [26]:
```
type(linhas_tb[2].get_text())
```

Out[26]:
```
str
```

In [27]:
```
prep_linha_v1(linhas_tb[2].get_text())
```

Out[27]:
```
'Estados Unidos\t33448422\t601079\t0'
```

Chamada com Tag

In [28]:
```
type(linhas_tb[2])
```

Out[28]:
```
bs4.element.Tag
```

In [29]:
```
prep_linha_v1(linhas_tb[2])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-29-8861c948244d> in <module>
----> 1 prep_linha_v1(linhas_tb[2])

<ipython-input-24-aaca6aa99a27> in prep_linha_v1(linha)
      2     # Remove os caracteres \n no começo e final da string
      3     # Em seguida, remove os espaços
----> 4     linha = linha.strip("\n ")
      5     linha = linha.replace(u"\xa0", u"")
      6     linha = re.sub("[\n|\t]+","\t", linha)

TypeError: 'NoneType' object is not callable
```

Investigando outros casos

Listando as transformações nos países.

In [30]:
```
for i in range(ini_tb,fim_tb+1):
    linha = linhas_tb[i].get_text()
```

```
    print(i, prep_linha_v1(linha))
```

```
2 Estados Unidos        33448422        601079  0
3 Índia 26752447        303720  23728011
4 Brasil        16720081        467706  15168330
5 França        5667324 109557  0
6 Turquia       5263697 47768   5131463
7 Rússia        5090249 122267  4702599
8 Reino Unido   4494699 127794  0
9 Itália        4220304 126221  3868332
10 Argentina    3852093 79320   3409253
11 Espanha      3687762 80049   0
12 Alemanha     3703807 89316   3498050
13 Colômbia     3432422 89297   3193406
14 Irã  2935443 80488   2494108
15 Polônia      2873527 73984   2641139
16 México       2420659 227840  1930608
17 Ucrânia      2206836 50857   2062572
18 Peru 1961087 69342   1914169
19 Chéquia      1662256 30126   1622432
20 África do Sul        1669231 56601   1563719
21 Países Baixos        1651780 17632   0
22 Indonésia    1 511 712       40 858  1 348 330
23 Canadá       1383214 25566   1326484
24 Chile        1394973 29385   1321600
25 Filipinas    1240716 21158   1167426
26 Iraque       1205522 16405   1120799
27 Romênia      1078142 30415   1040869
28 Suécia       1068473 14451   0
29 Bélgica      1063499 24968   0
30 Paquistão    924667  20930   848685
31 Portugal     850262  17026   810271
32 Israel       839508  6413    832693
33 Hungria      804382  29728   696029
34 Bangladesh   802305  12660   742151
35 Jordânia     737888  9489    718123
36 Japão        749130  13140   685365
37 Sérvia       712989  6881    0
38 Suíça        696213  10268   317600
39 Áustria      645552  10621   630274
40 Emirados Árabes Unidos       574958  1686    554589
41 Nepal        571111  7555    461563
42 Malásia      587165  2993    501898
43 Líbano       540630  7735    520473
44 Marrocos     519610  9154    507528
45 Arábia Saudita       451687  7377    434439
46 Equador      427690  20620   375151
47 Bulgária     418813  17726   383765
48 Grécia       404163  12122   data-sort-value="-1"
49 Bielorrússia 395990  2871    387338
50 Eslováquia   389990  12366   0
51 Cazaquistão  387672  3974    357826
52 Panamá       378828  6377    366039
53 Bolívia      374718  14639   297580
54 Croácia      356829  8042    346796
55 Paraguai     358244  9293    294994
56 Geórgia      346150  4834    329516
57 Tunísia      346986  12720   305055
58 Azerbaijão   334132  4921    325040
59 Costa Rica   321279  4074    244782
60 Palestina    308732  3503    300919
61 Kuwait       311846  1779    296242
62 República Dominicana 294021  3634    241806
63 Uruguai      298006  4342    257030
64 Dinamarca    283089  2516    267808
65 Lituânia     275198  4283    256887
66 Etiópia      271790  4171    239475
67 Irlanda      262319  4941    0
68 Egito        263606  15136   192823
```

69	Moldávia	255241	6114	247609	
70	Eslovênia	254419	4380	0	
71	Guatemala	257167	8214	235641	
72	Honduras	238820	6379	85279	
73	Venezuela	235567	2661	216746	
74	Bahrain	242790	1009	213827	
75	Armênia	222870	4446	213578	
76	Qatar	217882	562	214020	
77	Omã	218271	2356	200421	
78	Bósnia e Herzegovina	204172	9303	175704	
79	Líbia	186072	3127	172117	
80	Sri Lanka	192547	1566	160714	
81	Quênia	171226	3206	117039	
82	Nigéria	166535	2099	159935	
83	Macedônia do Norte	155304	5423	148798	
84	Tailândia	165462	1107	114578	
85	Myanmar	143823	3218	132388	
86	Coreia do Sul	141476	1965	132068	
87	Cuba	144514	977	137614	
88	Albânia	132351	2451	129521	
89	Letônia	133866	2386	126999	
90	Estônia	129804	1259	123589	
91	Argélia	128913	3472	89839	
92	Noruega	125764	785	88952	
93	Porto Rico	122097	2509	0	
94	Kosovo	107410	2243	102966	
95	Quirguistão	105469	1821	99175	
96	Montenegro	99683	1587	97351	
97	Uzbequistão	100726	691	96569	
98	Zâmbia	96563	1284	92108	
99	Gana	94011	785	92057	
100	Finlândia	92770	959	31000	
101	China (continental)	91146	4636	86164	
102	Camarões	78929	1275	57008	
103	El Salvador	73702	2252	68803	
104	Chipre	72515	360	0	
105	Moçambique	70923	836	69555	
106	Luxemburgo	70027	818	68316	
107	Afeganistão	75144	3034	57963	
108	Singapura	62100	33	61481	
109	Maldivas	64396	161	35595	
110	Mongólia	60372	286	51448	
111	Namíbia	56264	865	51048	
112	Botswana	54973	849	51259	
113	Jamaica	48594	949	25485	
114	Costa do Marfim	47319	306	46758	
115	Uganda	48676	364	46150	
116	Senegal	41494	1142	40146	
117	Madagáscar	41366	841	39101	
118	Zimbábue	39031	1599	36661	
119	Donetsk	39230	2965	31693	
120	Sudão	35656	2662	29364	
121	Malawi	34360	1156	32629	
122	Angola	34752	792	28190	
123	Rep. Dem. do Congo	31934	786	27666	
124	Malta	30553	419	30058	
125	Austrália	30130	910	0	
126	Cabo Verde	30694	266	28944	
127	Camboja	31460	230	24042	
128	Ruanda	27064	358	25948	
129	Síria	24495	1770	21604	
130	Gabão	24429	152	22118	
131	Guiné	23194	162	21193	
132	Trinidad e Tobago	24314	507	14249	
133	Mauritânia	19598	463	18582	
134	Polinésia Francesa	18879	142	18658	
135	Essuatíni	18618	673	17882	
136	Guiana	16952	389	14552	
137	Papua-Nova Guiné	15910	162	15067	

138	Abecásia	15360	234	14737		
139	Somália	14660	769	6764		
140	Mali	14281	517	9740		
141	Haiti	14931	321	12552		
142	Suriname	15128	313	11877		
143	Tajiquistão	13714	91	13218		
144	Andorra	13744	127	13507		
145	Burkina Faso	13435	167	13256		
146	Togo	13481	125	12950		
147	Belize	12819	325	12417		
148	Curaçao	12274	122	12132		
149	Hong Kong	11849	210	11561		
150	República do Congo	11845	154	8208		
151	Bahamas	11893	230	10903		
152	Djibouti	11542	154	11381		
153	Seicheles	11621	42	10499		
154	Aruba	11006	107	10845		
155	Lesoto	10831	326	6434		
156	Sudão do Sul	10688	115	10514		
157	Guiné Equatorial	8529	118	8146		
158	Guam	8187	139	7986		
159	Benim	8025	101	7893		
160	Nicarágua	7324	186	0		
161	Chipre do Norte	7345	33	6998		
162	República Centro-Africana	7091	98	5112		
163	Taiwan	9389	149	1133		
164	Iémen	6759	1323	3472		
165	Islândia	6595	30	6521		
166	Vietnã	7813	49	3085		
167	Gâmbia	5993	179	5780		
168	Timor-Leste	6897	16	4233		
169	Níger	5410	192	5083		
170	San Marino	5090	90	4998		
171	Santa Lúcia	5072	79	4826		
172	Chade	4935	173	4747		
173	Lugansk	4784	441	4172		
174	Burundi	4828	6	773		
175	Gibraltar	4298	94	4194		
176	Serra Leoa	4140	79	3128		
177	Eritreia	4145	14	3855		
178	Barbados	4017	47	3922		
179	Somalilândia	4608	311	3899		
180	Comores	3882	146	3719		
181	Guiné-Bissau	3766	68	3516		
182	Ilhas Virgens Americanas	3465	27	3313		
183	Ossétia do Sul	3343	60+	3198		
184	Jersey	3243	69	3179		
185	Liechtenstein	3016	58	2934		
186	Artsaque	2751	31	337		
187	Mônaco	2508	33	2467		
188	Bermudas	2494	33	2441		
189	Ilhas Turcas e Caicos	2417	17	2384		
190	São Martinho (Países Baixos)	2433	28	2334		
191	São Tomé e Príncipe	2345	38	2290		
192	Nova Zelândia	2323	26	2274		
193	Libéria	2219	86	2044		
194	Laos	1929	3	1599		
195	Ilha de Man	1594	29	1562		
196	Bonaire	1589	17	1549		
197	Butão	1639	1	1308		
198	Maurícia	1418	18	1198		
199	Antígua e Barbuda	1262	42	1213		
200	Guernsey	823	14	808		
201	Diamond Princess	712	14	698		
202	Tanzânia	0	0	0		
203	Ilhas Faroe	723	1	677		
204	Ilhas Cayman	584	2	571		
205	Fiji	508	4	181		
206	Ilhas Virgens Britânicas	292	1	285		

```
207 Brunei      244     3       231
208 Dominica    188     0       186
209 Ilhas Marianas Setentrionais        183     2       32
210 Granada     160     1       158
211 Costa Atlantica     148     0       148
212 Greg Mortimer       128     1       0
213 Nova Caledônia      128     0       30
214 Anguilla    109     0       109
215 São Vicente e Granadinas    98      0       81
216 Ilhas Malvinas      63      0       63
217 São Cristóvão e Neves       73      0       48
218 Macau       51      0       49
219 Gronelândia 40      0       34
220 Vaticano    29      0       27
221 Saint Pierre e Miquelon     25      0       25
222 Santo Eustáquio     20      0       20
223 Montserrat  20      1       18
224 MS Zaandam  13      4       0
225 Coral Princess      12      3       0
226 Saba        7       0       7
```

Pela Grécia, vemos que remos ter tratar um comando HTML em uma das colunas.

In [ ]:
```
prep_linha_v1(linhas_tb[48].get_text())
```

A Ossétia do Sul também necessita ser tratada.

In [31]:
```
prep_linha_v1(linhas_tb[183].get_text())
```

Out[31]:
```
'Ossétia do Sul\t3343\t60+\t3198'
```

In [32]:
```
def prep_linha_v2(linha):
    # Remove os caracteres \n no começo e final da string
    # Em seguida, remove os espaços
    linha = linha.strip("\n ")
    linha = linha.replace(u"\xa0", u"")
    linha = re.sub("[\n|\t]+","\t", linha)
    linha = re.sub(r"\[[\d|a-z]+\]", "", linha)
    linha = linha.strip("\t")
    linha = re.sub(r"—", "0", linha)

    # Trata a Grécia (parte por parte)
    linha = linha.replace("data-sort-value=\"-1\"", "0")

    # Trata a Ossétia do Sul
    linha = linha.replace("+", "")
    return linha
```

In [33]:
```
prep_linha_v2(linhas_tb[48].get_text())
```

Out[33]:
```
'Grécia\t404163\t12122\t0'
```

In [35]:
```
prep_linha_v2(linhas_tb[183].get_text())
```

Out[35]:
```
'Ossétia do Sul\t3343\t60\t3198'
```

Analisando a Indonésia, descobrimos mais uma surpresa.

In [36]:
```
prep_linha_v2(linhas_tb[22].get_text())
```

Out[36]:
```
'Indonésia\t1 511 712\t40 858\t1 348 330'
```

Para facilitar, vamos tratar este caso direto no Pandas.

Preparando o texto do arquivo final

Para criarmos o texto arquivo CSV final, basta fazer uma concatenação de todas as linhas da tabela, incluindo um `\n` ao final da linha.

In [37]:
```
# Esta primeira linha define o título da tabela
texto_final = "País\tCasos\tMortes\tCurados\n"
for i in range(ini_tb, fim_tb+1):
    # Preste atenção no .get_text(), que transforma a chamada em string
    linha = prep_linha_v2(linhas_tb[i].get_text())
    texto_final += linha
    if (i < fim_tb):
        texto_final += "\n"
print(texto_final)
```

```
País    Casos   Mortes  Curados
Estados Unidos  33448422        601079  0
Índia   26752447        303720  23728011
Brasil  16720081        467706  15168330
França  5667324 109557  0
Turquia 5263697 47768   5131463
Rússia  5090249 122267  4702599
Reino Unido     4494699 127794  0
Itália  4220304 126221  3868332
Argentina       3852093 79320   3409253
Espanha 3687762 80049   0
Alemanha        3703807 89316   3498050
Colômbia        3432422 89297   3193406
Irã     2935443 80488   2494108
Polônia 2873527 73984   2641139
México  2420659 227840  1930608
Ucrânia 2206836 50857   2062572
Peru    1961087 69342   1914169
Chéquia 1662256 30126   1622432
África do Sul   1669231 56601   1563719
Países Baixos   1651780 17632   0
Indonésia       1 511 712       40 858  1 348 330
Canadá  1383214 25566   1326484
Chile   1394973 29385   1321600
Filipinas       1240716 21158   1167426
Iraque  1205522 16405   1120799
Romênia 1078142 30415   1040869
Suécia  1068473 14451   0
Bélgica 1063499 24968   0
Paquistão       924667  20930   848685
Portugal        850262  17026   810271
Israel  839508  6413    832693
Hungria 804382  29728   696029
Bangladesh      802305  12660   742151
Jordânia        737888  9489    718123
Japão   749130  13140   685365
Sérvia  712989  6881    0
Suíça   696213  10268   317600
Áustria 645552  10621   630274
Emirados Árabes Unidos  574958  1686    554589
Nepal   571111  7555    461563
Malásia 587165  2993    501898
Líbano  540630  7735    520473
Marrocos        519610  9154    507528
Arábia Saudita  451687  7377    434439
Equador 427690  20620   375151
Bulgária        418813  17726   383765
Grécia  404163  12122   0
Bielorrússia    395990  2871    387338
Eslováquia      389990  12366   0
Cazaquistão     387672  3974    357826
Panamá  378828  6377    366039
Bolívia 374718  14639   297580
```

Croácia	356829	8042	346796	
Paraguai	358244	9293	294994	
Geórgia	346150	4834	329516	
Tunísia	346986	12720	305055	
Azerbaijão	334132	4921	325040	
Costa Rica	321279	4074	244782	
Palestina	308732	3503	300919	
Kuwait	311846	1779	296242	
República Dominicana	294021	3634	241806	
Uruguai	298006	4342	257030	
Dinamarca	283089	2516	267808	
Lituânia	275198	4283	256887	
Etiópia	271790	4171	239475	
Irlanda	262319	4941	0	
Egito	263606	15136	192823	
Moldávia	255241	6114	247609	
Eslovênia	254419	4380	0	
Guatemala	257167	8214	235641	
Honduras	238820	6379	85279	
Venezuela	235567	2661	216746	
Bahrain	242790	1009	213827	
Armênia	222870	4446	213578	
Qatar	217882	562	214020	
Omã	218271	2356	200421	
Bósnia e Herzegovina	204172	9303	175704	
Líbia	186072	3127	172117	
Sri Lanka	192547	1566	160714	
Quênia	171226	3206	117039	
Nigéria	166535	2099	159935	
Macedônia do Norte	155304	5423	148798	
Tailândia	165462	1107	114578	
Myanmar	143823	3218	132388	
Coreia do Sul	141476	1965	132068	
Cuba	144514	977	137614	
Albânia	132351	2451	129521	
Letônia	133866	2386	126999	
Estônia	129804	1259	123589	
Argélia	128913	3472	89839	
Noruega	125764	785	88952	
Porto Rico	122097	2509	0	
Kosovo	107410	2243	102966	
Quirguistão	105469	1821	99175	
Montenegro	99683	1587	97351	
Uzbequistão	100726	691	96569	
Zâmbia	96563	1284	92108	
Gana	94011	785	92057	
Finlândia	92770	959	31000	
China (continental)	91146	4636	86164	
Camarões	78929	1275	57008	
El Salvador	73702	2252	68803	
Chipre	72515	360	0	
Moçambique	70923	836	69555	
Luxemburgo	70027	818	68316	
Afeganistão	75144	3034	57963	
Singapura	62100	33	61481	
Maldivas	64396	161	35595	
Mongólia	60372	286	51448	
Namíbia	56264	865	51048	
Botswana	54973	849	51259	
Jamaica	48594	949	25485	
Costa do Marfim	47319	306	46758	
Uganda	48676	364	46150	
Senegal	41494	1142	40146	
Madagáscar	41366	841	39101	
Zimbábue	39031	1599	36661	
Donetsk	39230	2965	31693	
Sudão	35656	2662	29364	
Malawi	34360	1156	32629	
Angola	34752	792	28190	

Rep. Dem. do Congo	31934	786	27666		
Malta	30553	419	30058		
Austrália	30130	910	0		
Cabo Verde	30694	266	28944		
Camboja	31460	230	24042		
Ruanda	27064	358	25948		
Síria	24495	1770	21604		
Gabão	24429	152	22118		
Guiné	23194	162	21193		
Trinidad e Tobago	24314	507	14249		
Mauritânia	19598	463	18582		
Polinésia Francesa	18879	142	18658		
Essuatíni	18618	673	17882		
Guiana	16952	389	14552		
Papua-Nova Guiné	15910	162	15067		
Abecásia	15360	234	14737		
Somália	14660	769	6764		
Mali	14281	517	9740		
Haiti	14931	321	12552		
Suriname	15128	313	11877		
Tajiquistão	13714	91	13218		
Andorra	13744	127	13507		
Burkina Faso	13435	167	13256		
Togo	13481	125	12950		
Belize	12819	325	12417		
Curaçao	12274	122	12132		
Hong Kong	11849	210	11561		
República do Congo	11845	154	8208		
Bahamas	11893	230	10903		
Djibouti	11542	154	11381		
Seicheles	11621	42	10499		
Aruba	11006	107	10845		
Lesoto	10831	326	6434		
Sudão do Sul	10688	115	10514		
Guiné Equatorial	8529	118	8146		
Guam	8187	139	7986		
Benim	8025	101	7893		
Nicarágua	7324	186	0		
Chipre do Norte	7345	33	6998		
República Centro-Africana	7091	98	5112		
Taiwan	9389	149	1133		
Iémen	6759	1323	3472		
Islândia	6595	30	6521		
Vietnã	7813	49	3085		
Gâmbia	5993	179	5780		
Timor-Leste	6897	16	4233		
Níger	5410	192	5083		
San Marino	5090	90	4998		
Santa Lúcia	5072	79	4826		
Chade	4935	173	4747		
Lugansk	4784	441	4172		
Burundi	4828	6	773		
Gibraltar	4298	94	4194		
Serra Leoa	4140	79	3128		
Eritreia	4145	14	3855		
Barbados	4017	47	3922		
Somalilândia	4608	311	3899		
Comores	3882	146	3719		
Guiné-Bissau	3766	68	3516		
Ilhas Virgens Americanas	3465	27	3313		
Ossétia do Sul	3343	60	3198		
Jersey	3243	69	3179		
Liechtenstein	3016	58	2934		
Artsaque	2751	31	337		
Mônaco	2508	33	2467		
Bermudas	2494	33	2441		
Ilhas Turcas e Caicos	2417	17	2384		
São Martinho (Países Baixos)	2433	28	2334		
São Tomé e Príncipe	2345	38	2290		

```
Nova Zelândia   2323    26      2274
Libéria 2219    86      2044
Laos    1929    3       1599
Ilha de Man     1594    29      1562
Bonaire 1589    17      1549
Butão   1639    1       1308
Maurícia        1418    18      1198
Antígua e Barbuda       1262    42      1213
Guernsey        823     14      808
Diamond Princess        712     14      698
Tanzânia        0       0       0
Ilhas Faroe     723     1       677
Ilhas Cayman    584     2       571
Fiji    508     4       181
Ilhas Virgens Britânicas        292     1       285
Brunei  244     3       231
Dominica        188     0       186
Ilhas Marianas Setentrionais    183     2       32
Granada 160     1       158
Costa Atlantica 148     0       148
Greg Mortimer   128     1       0
Nova Caledônia  128     0       30
Anguilla        109     0       109
São Vicente e Granadinas        98      0       81
Ilhas Malvinas  63      0       63
São Cristóvão e Neves   73      0       48
Macau   51      0       49
Gronelândia     40      0       34
Vaticano        29      0       27
Saint Pierre e Miquelon 25      0       25
Santo Eustáquio 20      0       20
Montserrat      20      1       18
MS Zaandam      13      4       0
Coral Princess  12      3       0
Saba    7       0       7
```

Gravando o texto em um arquivo

Para se gravar o texto em um arquivo, basta utilizar o código a seguir.

In [38]:
```
arquivo = open("lista-paises.tsv", "w")
arquivo.write(texto_final)
arquivo.close()
```

Lendo o arquivo CSV com o Pandas

In [39]:
```
import pandas as pd
df = pd.read_csv("lista-paises.tsv", sep='\t')
df
```

Out[39]:

	País	Casos	Mortes	Curados
0	Estados Unidos	33448422	601079	0
1	Índia	26752447	303720	23728011
2	Brasil	16720081	467706	15168330
3	França	5667324	109557	0
4	Turquia	5263697	47768	5131463
...	...	...	...	...
220	Santo Eustáquio	20	0	20
221	Montserrat	20	1	18

	País	Casos	Mortes	Curados
222	MS Zaandam	13	4	0
223	Coral Princess	12	3	0
224	Saba	7	0	7

225 rows × 4 columns

Indonésia

In [40]:
```
df[df['País'] == 'Indonésia']
```

Out[40]:

	País	Casos	Mortes	Curados
20	Indonésia	1 511 712	40 858	1 348 330

A função lambda é uma função anônima (sem nome), em que o parâmetro é designado por celula. A função é aplicada sobre cada célula da coluna. Assim para cada célula, a função replace será executada.

In [41]:
```
df['Casos'] = df.Casos.apply(lambda celula: celula.replace(" ", ""))
df['Mortes'] = df.Mortes.apply(lambda celula: celula.replace(" ", ""))
df['Curados'] = df.Curados.apply(lambda celula: celula.replace(" ", ""))
```

In [42]:
```
df[df['País'] == 'Indonésia']
```

Out[42]:

	País	Casos	Mortes	Curados
20	Indonésia	1511712	40858	1348330

Convertendo para tipos numéricos

Se tudo correr bem nessa conversão, é porque tudo foi tratado adequadamente. Caso ocorra problemas aqui, você deve inspecionar a tabela para encontrar onde estão os problemas.

In [45]:
```
df['Casos'] = df.Casos.astype(int)
df['Mortes'] = df.Mortes.astype(int)
df['Curados'] = df.Curados.astype(int)
```

Tarefas

1) Utilizando o plotly, faça a análise da evolução de Casos, Mortes e Curados do Dataset.

2) Calcule a estatística descritiva (média, soma, mediana e desvio-padrão) dos países do dataset.

3) Encontre uma tabela na Wikipedia do seu interesse e aplique as técnicas apresentadas neste caderno para criar um DataFrame, criar gráficos no Plotly e fazer as estatísticas descritivas.