Nome: João Emanuel da S - Matricula: 162080263 - Data 18/agosto/2020

Dataframes

```
In [1]:
#!pip install pandas
#!conda install pandas
In [2]:
import pandas as pd
In [3]:
df = pd.DataFrame()
df
Out[3]:
In [4]:
type(df)
Out[4]:
pandas.core.frame.DataFrame
In [5]:
df['Nome'] = [ 'Jessica', 'Aline']
df
Out[5]:
    Nome
0 Jessica
1
    Aline
```

```
In [6]:

df['cre'] = [ 7.8, 8.3 ]

Out[6]:

    Nome cre
    O Jessica 7.8
    1 Aline 8.3
```

Dataframe de casos COVID-19 - em 13/agosto/2020

```
In [7]:
```

```
atributos = [ 'Local', 'Confirmados', 'Novos casos (60 dias)', 'Casos (milhão)',
'recuperados', 'Mortes']
```

In [8]:

```
atributos
```

```
Out[8]:
['Local',
  'Confirmados',
  'Novos casos (60 dias)',
  'Casos (milhão)',
  'recuperados',
  'Mortes']
```

In [9]:

Out[9]:

```
{'Estado': ['São Paulo', 'Bahia', 'Ceará', 'Rio de Janeiro', 'Paraíb
a'],
'Confirmados': [655181, 203020, 192422, 185610, 92897],
'Casos (milhão)': [14879, 13421, 21760, 11276, 23555],
'Mortes': [25869, 4135, 8052, 14295, 2071]}
```

In [10]:

```
df = pd.DataFrame(data)
df
```

Out[10]:

	Estado	Confirmados	Casos (milhão)	Mortes
0	São Paulo	655181	14879	25869
1	Bahia	203020	13421	4135
2	Ceará	192422	21760	8052
3	Rio de Janeiro	185610	11276	14295
4	Paraíba	92897	23555	2071

In [11]:

```
type(df)
```

Out[11]:

pandas.core.frame.DataFrame

In [12]:

```
df['Estado']
```

Out[12]:

```
0 São Paulo
1 Bahia
2 Ceará
3 Rio de Janeiro
4 Paraíba
```

Name: Estado, dtype: object

In [13]:

```
df[ ['Estado', 'Confirmados']]
```

Out[13]:

	Estado	Confirmados
0	São Paulo	655181
1	Bahia	203020
2	Ceará	192422
3	Rio de Janeiro	185610
4	Paraíba	92897

In [14]:

df

Out[14]:

	Estado	Confirmados	Casos (milhão)	Mortes
0	São Paulo	655181	14879	25869
1	Bahia	203020	13421	4135
2	Ceará	192422	21760	8052
3	Rio de Janeiro	185610	11276	14295
4	Paraíba	92897	23555	2071

In [15]:

df.head()

Out[15]:

	Estado	Confirmados	Casos (milhão)	Mortes
0	São Paulo	655181	14879	25869
1	Bahia	203020	13421	4135
2	Ceará	192422	21760	8052
3	Rio de Janeiro	185610	11276	14295
4	Paraíba	92897	23555	2071

In [16]:

df.tail()

Out[16]:

	Estado	Confirmados	Casos (milhão)	Mortes
0	São Paulo	655181	14879	25869
1	Bahia	203020	13421	4135
2	Ceará	192422	21760	8052
3	Rio de Janeiro	185610	11276	14295
4	Paraíba	92897	23555	2071

In [17]:

list(df.index)

Out[17]:

[0, 1, 2, 3, 4]

```
In [18]:
```

```
atributo = df.columns.values
atributo
```

Out[18]:

```
array(['Estado', 'Confirmados', 'Casos (milhão)', 'Mortes'], dtype=o
bject)
```

In [19]:

```
atributo[0]
```

Out[19]:

'Estado'

In [20]:

```
df.values
```

Out[20]:

In [21]:

```
df2 = df
del df2['Estado']
df2
```

Out[21]:

	Confirmados	Casos (milnao)	Mortes
0	655181	14879	25869
1	203020	13421	4135
2	192422	21760	8052
3	185610	11276	14295
4	92897	23555	2071

In [22]:

```
k = df2.values
print(k)
```

```
[[655181 14879 25869]
[203020 13421 4135]
[192422 21760 8052]
[185610 11276 14295]
[ 92897 23555 2071]]
```

```
In [23]:
```

```
x = df[ ['Confirmados', 'Casos (milhão)']]
x
```

Out[23]:

	Confirmados	Casos (milhão)
0	655181	14879
1	203020	13421
2	192422	21760
3	185610	11276
4	92897	23555
In	[24]:	

```
y = df['Mortes']
y
```

Out[24]:

0 25869 1 4135 2 8052 3 14295

2071

Name: Mortes, dtype: int64

In [25]:

4

```
x = x.values
x
```

Out[25]:

In [26]:

```
y = y.values
y
```

Out[26]:

```
array([25869, 4135, 8052, 14295, 2071], dtype=int64)
```

In [27]:

```
from sklearn import linear_model
from sklearn.metrics import r2_score
```

```
In [28]:
```

```
# Criando e treinando um modelo
modelo = linear_model.LinearRegression()
X = x
modelo.fit(X,y)
```

Out[28]:

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, norma
lize=False)

In [29]:

```
def r2_est(X,y):
    modelo = linear_model.LinearRegression(normalize = False, fit_intercept = Tr
ue)
    return r2_score(y, modelo.fit(X,y).predict(X))
```

In [30]:

```
print ('R2: %0.3f' % r2_est(X,y))
```

R2: 0.854

In [31]:

df

Out[31]:

	Confirmados	Casos (milhão)	Mortes
0	655181	14879	25869
1	203020	13421	4135
2	192422	21760	8052
3	185610	11276	14295
4	92897	23555	2071

In [32]:

```
import numpy as np
z = np.array([100000, 15000])
int(modelo.predict(z.reshape(1, -1)))
```

Out[32]:

```
In [33]:
```

```
z = np.array([150000, 25000])
int(modelo.predict(z.reshape(1, -1)))
```

Out[33]:

3946

In [34]:

df

Out[34]:

	Confirmados	Casos (milhão)	Mortes
0	655181	14879	25869
1	203020	13421	4135
2	192422	21760	8052
3	185610	11276	14295
4	92897	23555	2071

Slicing

In [35]:

df[1:]

Out[35]:

	Confirmados	Casos (milhão)	Mortes
1	203020	13421	4135
2	192422	21760	8052
3	185610	11276	14295
4	92897	23555	2071

In [36]:

df[:3]

Out[36]:

	Confirmados	Casos (milhão)	Mortes
0	655181	14879	25869
1	203020	13421	4135
2	192422	21760	8052

In [37]:

df[1:4] # 4 é exclusive = 4 - 1

Out[37]:

	Confirmados	Casos (milhão)	Mortes
1	203020	13421	4135
2	192422	21760	8052
3	185610	11276	14295

In [38]:

df.describe()

Out[38]:

	Confirmados	Casos (milhão)	Mortes
count	5.000000	5.000000	5.00000
mean	265826.000000	16978.200000	10884.40000
std	222074.932497	5378.087643	9584.14497
min	92897.000000	11276.000000	2071.00000
25%	185610.000000	13421.000000	4135.00000
50%	192422.000000	14879.000000	8052.00000
75%	203020.000000	21760.000000	14295.00000
max	655181.000000	23555.000000	25869.00000

In [39]:

df.describe().T

Out[39]:

	count	mean	std	min	25%	50%	75%	ma
Confirmados	5.0	265826.0	222074.932497	92897.0	185610.0	192422.0	203020.0	655181.
Casos (milhão)	5.0	16978.2	5378.087643	11276.0	13421.0	14879.0	21760.0	23555.
Mortes	5.0	10884.4	9584.144970	2071.0	4135.0	8052.0	14295.0	25869.

In [40]:

outliers - Valores Fora do intervalo

```
In [44]:
df.min()
Out[44]:
Confirmados
                   92897
Casos (milhão)
                   11276
Mortes
                    2071
dtype: int64
In [45]:
df.max()
Out[45]:
Confirmados
                   655181
Casos (milhão)
                    23555
Mortes
                    25869
dtype: int64
In [46]:
df['Casos (milhão)'].min()
Out[46]:
11276
In [47]:
df.Confirmados.min()
Out[47]:
92897
```

Leitura de Datasets usando a biblioteca Pandas (Dataframe)

```
In [48]:
#!dir
!ls

'ls' nÆo , reconhecido como um comando interno
ou externo, um programa oper vel ou um arquivo em lotes.

In [49]:
import pandas as pd
```

```
In [50]:
```

```
url = 'caso_full.csv'
df = pd.read_csv(url)
len(df)
```

Out[50]:

554845

In [51]:

df.head()

Out[51]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last	is
0	São Paulo	3550308.0	2020- 02-25	9	12252023.0	False	
1	NaN	35.0	2020- 02-25	9	45919049.0	False	
2	São Paulo	3550308.0	2020- 02-26	9	12252023.0	False	
3	NaN	35.0	2020- 02-26	9	45919049.0	False	
4	São Paulo	3550308.0	2020- 02-27	9	12252023.0	False	
4							•

In [52]:

Estrutura do Dataframe df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 554845 entries, 0 to 554844
Data columns (total 17 columns):
                                                  550575 non-null obj
citv
ect
city_ibge_code
                                                  552352 non-null flo
at64
                                                  554845 non-null obi
date
ect
                                                  554845 non-null int
epidemiological week
64
estimated population 2019
                                                  552352 non-null flo
at64
is last
                                                  554845 non-null boo
                                                  554845 non-null boo
is repeated
                                                  554845 non-null int
last available confirmed
last available confirmed per 100k inhabitants
                                                  540486 non-null flo
last available date
                                                  554845 non-null obj
ect
                                                  554845 non-null flo
last available death rate
at64
last available deaths
                                                  554845 non-null int
                                                  554845 non-null int
order for place
                                                  554845 non-null obj
place_type
ect
state
                                                  554845 non-null obj
ect
                                                  554845 non-null int
new confirmed
new_deaths
                                                  554845 non-null int
dtypes: bool(2), float64(4), int64(6), object(5)
memory usage: 64.6+ MB
```

```
In [53]:
```

```
df.head()
```

Out[53]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last	is
0	São Paulo	3550308.0	2020- 02-25	9	12252023.0	False	
1	NaN	35.0	2020- 02-25	9	45919049.0	False	
2	São Paulo	3550308.0	2020- 02-26	9	12252023.0	False	
3	NaN	35.0	2020- 02-26	9	45919049.0	False	
4	São Paulo	3550308.0	2020- 02-27	9	12252023.0	False	

In [54]:

4

```
df.epidemiological_week.unique()
```

Out[54]:

```
array([ 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2 4, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34], dtype=int64)
```

In [55]:

len(df)

Out[55]:

554845

filtrando dados

In [56]:

```
# Semana Epidemiológica = 34 <== Modificar aqui (entre 9 e 33)
df34 = df[ df.epidemiological_week == 34 ]
len(df34)</pre>
```

Out[56]:

11137

In [57]:

```
pd.options.display.float_format = '{:.0f}'.format
```

In [58]:

df34.describe().T

Out[58]:

	count	mean	std	min	25%	50%
city_ibge_code	11099	3235224	1009762	11	2509503	3145208
epidemiological_week	11137	34	0	34	34	34
estimated_population_2019	11099	75702	860638	781	5529	11819
last_available_confirmed	11137	1202	12955	0	30	9(
last_available_confirmed_per_100k_inhabitants	11072	1209	1204	7	418	876
last_available_death_rate	11137	0	0	0	0	(
last_available_deaths	11137	39	510	0	0	1
order_for_place	11137	99	29	1	81	101
new_confirmed	11137	7	91	-3048	0	(
new_deaths	11137	0	3	-99	0	(
4						>

In [59]:

len(df34.city.unique())

Out[59]:

5256

In [60]:

df34.head()

Out[60]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019
543708	Acrelândia	1200013	2020- 08-16	34	15256
543709	Assis Brasil	1200054	2020- 08-16	34	7417
543710	Brasiléia	1200104	2020- 08-16	34	26278
543711	Bujari	1200138	2020- 08-16	34	10266
543712	Capixaba	1200179	2020- 08-16	34	11733
4					>

```
In [61]:
```

```
df34.date.unique()
```

Out[61]:

array(['2020-08-16', '2020-08-17'], dtype=object)

In [62]:

len(df34)

Out[62]:

11137

In [63]:

```
# data = '2020-08-17'
df16 = df34[ df.date == '2020-08-17'].reset_index(drop=True)
len(df16)
```

C:\Users\djylm\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
UserWarning: Boolean Series key will be reindexed to match DataFrame
index.

Out[63]:

5569

In [64]:

df16.head()

Out[64]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_las
0	Acrelândia	1200013	2020- 08-17	34	15256	False
1	Assis Brasil	1200054	2020- 08-17	34	7417	False
2	Brasiléia	1200104	2020- 08-17	34	26278	False
3	Bujari	1200138	2020- 08-17	34	10266	False
4	Capixaba	1200179	2020- 08-17	34	11733	False
4						•

In [65]:

```
df16.city.value_counts()
```

Out[65]:

Importados/Indefinidos 19 Bom Jesus 5 Vera Cruz 4 São Francisco 4 Santa Luzia 4 . . São Pedro do Iguaçu 1 Santa Cruz do Sul 1 Divisa Alegre 1 Jaci 1 Cambuquira 1

Name: city, Length: 5255, dtype: int64

In [67]:

df16.head()

Out[67]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_las
0	Acrelândia	1200013	2020- 08-17	34	15256	False
1	Assis Brasil	1200054	2020- 08-17	34	7417	False
2	Brasiléia	1200104	2020- 08-17	34	26278	False
3	Bujari	1200138	2020- 08-17	34	10266	False
4	Capixaba	1200179	2020- 08-17	34	11733	False
4						•

In [68]:

df_novos_casos = df16[df16.new_confirmed > 0].reset_index(drop=True)
len(df_novos_casos)

Out[68]:

In [69]:

df_novos_casos.head()

Out[69]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_las
0	Arapiraca	2700300	2020- 08-17	34	231747	Tru
1	Atalaia	2700409	2020- 08-17	34	47185	Tru
2	Barra de São Miguel	2700607	2020- 08-17	34	8322	Tru
3	Cajueiro	2701308	2020- 08-17	34	21264	Tru
4	Campestre	2701357	2020- 08-17	34	6936	Tru

In [70]:

ordenar do maior para o menor = last_available_confirmed

dfo = df_novos_casos.sort_values(by='last_available_confirmed', ascending=False)
.reset_index(drop=True)
len(dfo)

Out[70]:

In [71]:

dfo.head(20)

Out[71]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last
0	NaN	35	2020- 08-17	34	45919049	True
1	São Paulo	3550308	2020- 08-17	34	12252023	True
2	NaN	29	2020- 08-17	34	14873064	True
3	NaN	23	2020- 08-17	34	9132078	True
4	NaN	33	2020- 08-17	34	17264943	True
5	NaN	15	2020- 08-17	34	8602865	True
6	NaN	31	2020- 08-17	34	21168791	True
7	NaN	53	2020- 08-17	34	3015268	True
8	NaN	21	2020- 08-17	34	7075181	True
9	NaN	42	2020- 08-17	34	7164788	True
10	NaN	26	2020- 08-17	34	9557071	True
11	NaN	13	2020- 08-17	34	4144597	True
12	NaN	41	2020- 08-17	34	11433957	True
13	NaN	52	2020- 08-17	34	7018354	True
14	NaN	32	2020- 08-17	34	4018650	True
15	NaN	43	2020- 08-17	34	11377239	True
16	NaN	25	2020- 08-17	34	4018127	True
17	Rio de Janeiro	3304557	2020- 08-17	34	6718903	True
18	NaN	51	2020- 08-17	34	3484466	True
19	NaN	27	2020- 08-17	34	3337357	True

```
In [72]:
list(dfo.place_type.unique())
Out[72]:
['state', 'city']
In [73]:
len(dfo)
Out[73]:
1880
In [74]:
df3 = dfo[ dfo.place_type == 'city' ].reset_index(drop=True)
len(df3)
Out[74]:
1857
In [75]:
1880 - 1857
Out[75]:
23
```

In [76]:

df3.head(10)

Out[76]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last
0	São Paulo	3550308	2020- 08-17	34	12252023	True
1	Rio de Janeiro	3304557	2020- 08-17	34	6718903	True
2	Salvador	2927408	2020- 08-17	34	2872347	True
3	Fortaleza	2304400	2020- 08-17	34	2669342	True
4	Belém	1501402	2020- 08-17	34	1492745	True
5	Belo Horizonte	3106200	2020- 08-17	34	2512070	True
6	Goiânia	5208707	2020- 08-17	34	1516113	True
7	Maceió	2704302	2020- 08-17	34	1018948	True
8	João Pessoa	2507507	2020- 08-17	34	809015	True
9	Curitiba	4106902	2020- 08-17	34	1933105	True

In [77]:

```
\label{eq:dfo} $$ dfo = df3.sort_values(by='last_available_confirmed', ascending=False).reset_inde $$ x(drop=True)$ $$ dfo.head(20)$
```

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is _.
0	São Paulo	3550308	2020- 08-17	34	12252023	
1	Rio de Janeiro	3304557	2020- 08-17	34	6718903	
2	Salvador	2927408	2020- 08-17	34	2872347	
3	Fortaleza	2304400	2020- 08-17	34	2669342	
4	Belém	1501402	2020- 08-17	34	1492745	
5	Belo Horizonte	3106200	2020- 08-17	34	2512070	
6	Goiânia	5208707	2020- 08-17	34	1516113	
7	Maceió	2704302	2020- 08-17	34	1018948	
8	João Pessoa	2507507	2020- 08-17	34	809015	
9	Curitiba	4106902	2020- 08-17	34	1933105	
10	Campinas	3509502	2020- 08-17	34	1204073	
11	Teresina	2211001	2020- 08-17	34	864845	
12	São Bernardo do Campo	3548708	2020- 08-17	34	838936	
13	Santos	3548500	2020- 08-17	34	433311	
14	Parauapebas	1505536	2020- 08-17	34	208273	
15	São Luís	2111300	2020- 08-17	34	1101884	
16	Uberlândia	3170206	2020- 08-17	34	691305	
17	Macapá	1600303	2020- 08-17	34	503327	
18	Aparecida de Goiânia	5201405	2020- 08-17	34	578179	
19	Guarulhos	3518800	2020- 08-17	34	1379182	

◆

In [78]:

```
dfo[ dfo.city == 'Campina Grande']
```

Out[78]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last
25	Campina Grande	2504009	2020- 08-17	34	409731	True
4						>

In [79]:

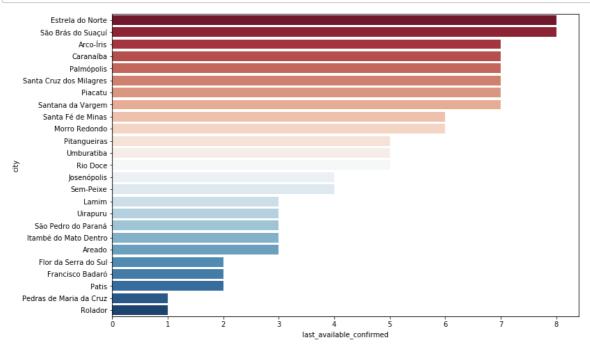
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))

ax = sns.barplot(x = 'last_available_confirmed', y = 'city', data = dfo[:25], pa lette='RdBu')
```

In [80]:

```
plt.figure(figsize=(12,8))
ax = sns.barplot(x = 'last_available_confirmed', y = 'city', data = dfo[-25:], p
alette='RdBu')
```



Exercício: Mudar a Semana Epidemiológica

A semana escolhida foi a semana 23 para se realizar a análise

In [81]:

```
# Semana Epidemiológica = 34 <== Modificar aqui (entre 9 e 33)

df23 = df[ df.epidemiological_week == 23 ]
len(df23)</pre>
```

Out[81]:

30922

In [82]:

```
pd.options.display.float_format = '{:.0f}'.format
```

In [83]:

df23.describe().T

Out[83]:

	count	mean	std	min	25%	50%
city_ibge_code	30803	3114982	1014043	11	2406205	3117900
epidemiological_week	30922	23	0	23	23	23
estimated_population_2019	30803	93656	965658	837	7342	15378
last_available_confirmed	30922	268	2982	0	3	10
last_available_confirmed_per_100k_inhabitants		191	337	1	30	78
last_available_death_rate	30922	0	0	0	0	0
last_available_deaths	30922	15	210	0	0	0
order_for_place	30922	36	21	1	19	34
new_confirmed	30922	11	125	-115	0	0
new_deaths	30922	0	7	-2	0	0
4						>

In [85]:

len(df23.city.unique())

Out[85]:

In [86]:

```
df23.head()
```

Out[86]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019
141225	Acrelândia	1200013	2020- 05-31	23	15256
141226	Assis Brasil	1200054	2020- 05-31	23	7417
141227	Brasiléia	1200104	2020- 05-31	23	26278
141228	Bujari	1200138	2020- 05-31	23	10266
141229	Capixaba	1200179	2020- 05-31	23	11733

In [87]:

```
df23.date.unique()
```

Out[87]:

```
array(['2020-05-31', '2020-06-01', '2020-06-02', '2020-06-03', '2020-06-04', '2020-06-05', '2020-06-06'], dtype=object)
```

In [88]:

```
len(df23)
```

Out[88]:

30922

In [90]:

```
# data = '2020-05-31'
df = df23[ df.date == '2020-05-31'].reset_index(drop=True)
len(df16)
```

C:\Users\djylm\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
UserWarning: Boolean Series key will be reindexed to match DataFrame
index.

Out[90]:

In [92]:

df.head()

Out[92]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last	is
0	São Paulo	3550308	2020- 02-25	9	12252023	False	
1	NaN	35	2020- 02-25	9	45919049	False	
2	São Paulo	3550308	2020- 02-26	9	12252023	False	
3	NaN	35	2020- 02-26	9	45919049	False	
4	São Paulo	3550308	2020- 02-27	9	12252023	False	

In [93]:

df.city.value_counts()

Out[93]:

Importados/Indefinidos	2493
Bom Jesus	517
Santa Luzia	486
São Domingos	429
Bonito	412
Arco-Íris	5
Porto Lucena	5
Nova Ramada	3
Amaral Ferrador	2
Rolador	1

Name: city, Length: 5255, dtype: int64

In [94]:

df.head()

Out[94]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last	is
0	São Paulo	3550308	2020- 02-25	9	12252023	False	
1	NaN	35	2020- 02-25	9	45919049	False	
2	São Paulo	3550308	2020- 02-26	9	12252023	False	
3	NaN	35	2020- 02-26	9	45919049	False	
4	São Paulo	3550308	2020- 02-27	9	12252023	False	
4							•

In [95]:

```
df_novos_casos = df[ df.new_confirmed > 0].reset_index(drop=True)
len(df_novos_casos)
```

Out[95]:

230398

In [96]:

```
df_novos_casos.head()
```

Out[96]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last	is
0	São Paulo	3550308	2020- 02-25	9	12252023	False	
1	NaN	35	2020- 02-25	9	45919049	False	
2	São Paulo	3550308	2020- 02-28	9	12252023	False	
3	NaN	35	2020- 02-28	9	45919049	False	
4	São Paulo	3550308	2020- 03-04	10	12252023	False	
4							•

```
In [97]:
# ordenar do maior para o menor = last_available_confirmed
dfo = df_novos_casos.sort_values(by='last_available_confirmed', ascending=False)
.reset index(drop=True)
len(dfo)
Out[97]:
230398
In [98]:
list(dfo.place type.unique())
Out[98]:
['state', 'city']
In [99]:
len(dfo)
Out[99]:
230398
In [100]:
df3 = dfo[ dfo.place_type == 'city' ].reset_index(drop=True)
len(df3)
Out[100]:
226353
In [101]:
1880 - 1857
Out[101]:
```

In [102]:

df3.head(10)

Out[102]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last	is
0	São Paulo	3550308	2020- 08-17	34	12252023	True	
1	São Paulo	3550308	2020- 08-15	33	12252023	False	
2	São Paulo	3550308	2020- 08-14	33	12252023	False	
3	São Paulo	3550308	2020- 08-13	33	12252023	False	
4	São Paulo	3550308	2020- 08-12	33	12252023	False	
5	São Paulo	3550308	2020- 08-11	33	12252023	False	
6	São Paulo	3550308	2020- 08-10	33	12252023	False	
7	São Paulo	3550308	2020- 08-09	33	12252023	False	
8	São Paulo	3550308	2020- 08-08	32	12252023	False	
9	São Paulo	3550308	2020- 08-07	32	12252023	False	

In [103]:

```
dfo = df3.sort_values(by='last_available_confirmed', ascending=False).reset_inde
x(drop=True)
dfo.head(20)
```

Out[103]:

	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is_last i
0	São Paulo	3550308	2020- 08-17	34	12252023	True
1	São Paulo	3550308	2020- 08-15	33	12252023	False
2	São Paulo	3550308	2020- 08-14	33	12252023	False
3	São Paulo	3550308	2020- 08-13	33	12252023	False
4	São Paulo	3550308	2020- 08-12	33	12252023	False
5	São Paulo	3550308	2020- 08-11	33	12252023	False
6	São Paulo	3550308	2020- 08-10	33	12252023	False
7	São Paulo	3550308	2020- 08-09	33	12252023	False
8	São Paulo	3550308	2020- 08-08	32	12252023	False
9	São Paulo	3550308	2020- 08-07	32	12252023	False
10	São Paulo	3550308	2020- 08-06	32	12252023	False
11	São Paulo	3550308	2020- 08-05	32	12252023	False
12	São Paulo	3550308	2020- 08-04	32	12252023	False
13	São Paulo	3550308	2020- 08-03	32	12252023	False
14	São Paulo	3550308	2020- 08-02	32	12252023	False
15	São Paulo	3550308	2020- 08-01	31	12252023	False
16	São Paulo	3550308	2020- 07-31	31	12252023	False
17	São Paulo	3550308	2020- 07-30	31	12252023	False
18	São Paulo	3550308	2020- 07-29	31	12252023	False
19	São Paulo	3550308	2020- 07-28	31	12252023	False

In [104]:

dfo[dfo.city == 'Campina Grande']

Out[104]:

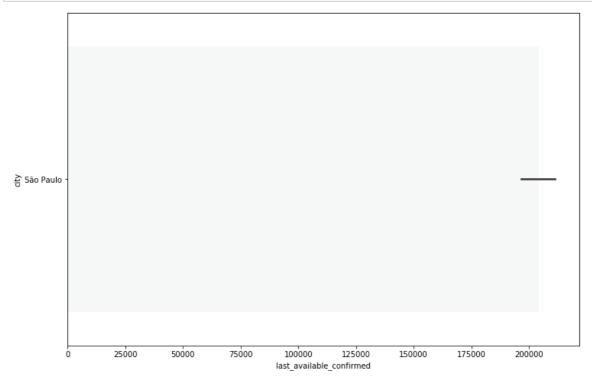
	city	city_ibge_code	date	epidemiological_week	estimated_population_2019	is		
1518	Campina Grande	2504009	2020- 08-17	34	409731			
1521	Campina Grande	2504009	2020- 08-16	34	409731			
1524	Campina Grande	2504009	2020- 08-15	33	409731			
1540	Campina Grande	2504009	2020- 08-14	33	409731			
1551	Campina Grande	2504009	2020- 08-13	33	409731			
201676	Campina Grande	2504009	2020- 04-16	16	409731			
213637	Campina Grande	2504009	2020- 04-15	16	409731			
215178	Campina Grande	2504009	2020- 04-05	15	409731			
219635	Campina Grande	2504009	2020- 03-28	13	409731			
226007	Campina Grande	2504009	2020- 03-27	13	409731			
119 rows	119 rows × 17 columns							

In [105]:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))

ax = sns.barplot(x = 'last_available_confirmed', y = 'city', data = dfo[:25], pa lette='RdBu')
```



In [106]:

```
plt.figure(figsize=(12,8))
ax = sns.barplot(x = 'last_available_confirmed', y = 'city', data = dfo[-25:], p
alette='RdBu')
```

