Nome: João Emanuel - Matricula: 162080263 - Data 25/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 ]
df
```

Out[6]:

	Nome	cre
0	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]:

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

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 [42]:

```
df.head()
```

Out[42]:

	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 [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 [43]:

```
list(df.index)
```

Out[43]:

```
[0, 1, 2, 3, 4]
```

In [44]:

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

Out[44]:

```
array(['Confirmados', 'Casos (milhão)', 'Mortes'], dtype=object)
```

In [19]:

```
atributo[0]
```

Out[19]:

```
'Estado'
```

In [20]:

```
df.values
```

Out[20]:

```
array([['São Paulo', 655181, 14879, 25869],
       ['Bahia', 203020, 13421, 4135],
       ['Ceará', 192422, 21760, 8052],
       ['Rio de Janeiro', 185610, 11276, 14295],
       ['Paraíba', 92897, 23555, 2071]], dtype=object)
```

In [21]:

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

Out[21]:

	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 [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
4     2071
Name: Mortes, dtype: int64
```

In [25]:

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

Out[25]:

```
array([[655181,  14879],
       [203020,  13421],
       [192422,  21760],
       [185610,  11276],
       [ 92897,  23555]])
```

In [26]:

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

Out[26]:

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

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()
```

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 [45]:

```
import numpy as np

z = np.array([100000, 15000])

int(modelo.predict(z.reshape(1, -1)))
```

Out[45]:

```
5536
```

In [46]:

```
z = np.array([150000, 25000])

int(modelo.predict(z.reshape(1, -1)))
```

Out[46]:

```
3946
```

In [33]:

```
df
```

Out[33]:

	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 [34]:

```
df[1:]
```

Out[34]:

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

In [35]:

```
df[:3]
```

Out[35]:

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

In [47]:

```
df[1:4] # 4 é exclusive = 4 - 1
```

Out[47]:

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

In [48]:

```
df.describe()
```

Out[48]:

	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 [49]:

```
df.describe().T
```

Out[49]:

	count	mean	std	min	25%	50%	75%	m
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	2355
Mortes	5.0	10884.4	9584.144970	2071.0	4135.0	8052.0	14295.0	2586

In [50]:

```
# outliers - Valores Fora do intervalo
```

In [37]:

```
df.min()
```

Out[37]:

```
Confirmados       92897
Casos (milhão)    11276
Mortes             2071
dtype: int64
```

In [38]:

```
df.max()
```

Out[38]:

```
Confirmados       655181
Casos (milhão)     23555
Mortes             25869
dtype: int64
```

In [39]:

```
df['Casos (milhão)'].min()
```

Out[39]:

```
11276
```

In [40]:

```
df.Confirmados.min()
```

Out[40]:

```
92897
```

Leitura de Datasets usando a biblioteca Pandas (Dataframe)

In [52]:

```
#!dir
!ls
```

```
Aula-02-Listas-Dicionarios-Python-11-Agosto-2020.ipynb
Aula-02-Python-11-Agosto-2020-pdf.pdf
Aula-02-Python-11-Agosto-2020-png.png
Aula-03-Dataframes-Python-13-Agosto-2020.ipynb
Aula-Python-01-Data-06-Agosto-2020 (7).pdf
Aula-Python-01-Data-06-Agosto-2020.ipynb
Aula-Python-01-Data-06-Agosto-2020.png
Aula-TEBD-Agosto-4-2020.xlsx
Untitled.ipynb
Untitled1.ipynb
Untitled2.ipynb
Untitled3.ipynb
Untitled4.ipynb
caso_full.csv
```

In [53]:

```
import pandas as pd
```

In [54]:

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

Out[54]:

```
554845
```

In [55]:

```
df.head()
```

Out[55]:

	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 [56]:

```
# Estrutura do Dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 554845 entries, 0 to 554844
Data columns (total 17 columns):
 #   Column                                         Non-Null Count
Dtype
---  ------                                         --------------
-----
 0   city                                           550575 non-null
object
 1   city_ibge_code                                 552352 non-null
float64
 2   date                                           554845 non-null
object
 3   epidemiological_week                           554845 non-null
int64
 4   estimated_population_2019                      552352 non-null
float64
 5   is_last                                        554845 non-null
bool
 6   is_repeated                                    554845 non-null
bool
 7   last_available_confirmed                       554845 non-null
int64
 8   last_available_confirmed_per_100k_inhabitants  540486 non-null
float64
 9   last_available_date                            554845 non-null
object
 10  last_available_death_rate                      554845 non-null
float64
 11  last_available_deaths                          554845 non-null
int64
 12  order_for_place                                554845 non-null
int64
 13  place_type                                     554845 non-null
object
 14  state                                          554845 non-null
object
 15  new_confirmed                                  554845 non-null
int64
 16  new_deaths                                     554845 non-null
int64
dtypes: bool(2), float64(4), int64(6), object(5)
memory usage: 64.6+ MB
```

In [57]:

```
df.head()
```

Out[57]:

	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 [58]:

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

Out[58]:

```
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])
```

In [59]:

```
len(df)
```

Out[59]:

```
554845
```

filtrando dados

In [61]:

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

Out[61]:

```
11137
```

In [65]:

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

In [66]:

```
df34.describe().T
```

Out[66]:

	count	mean	std	min	25%	
city_ibge_code	11099	3235224	1009762	11	2509503	3145
epidemiological_week	11137	34	0	34	34	
estimated_population_2019	11099	75702	860638	781	5529	11
last_available_confirmed	11137	1202	12955	0	30	
last_available_confirmed_per_100k_inhabitants	11072	1209	1204	7	418	
last_available_death_rate	11137	0	0	0	0	
last_available_deaths	11137	39	510	0	0	
order_for_place	11137	99	29	1	81	
new_confirmed	11137	7	91	-3048	0	
new_deaths	11137	0	3	-99	0	

In [69]:

```
len(df34.city.unique())
```

Out[69]:

```
5256
```

In [70]:

```
df34.head()
```

Out[70]:

	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

In [71]:

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

Out[71]:

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

In [72]:

```
len(df34)
```

Out[72]:

```
11137
```

In [75]:

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

```
<ipython-input-75-8024aa1bded5>:2: UserWarning: Boolean Series key w
ill be reindexed to match DataFrame index.
  df16 = df34[ df.date == '2020-08-17'].reset_index(drop=True)
```

Out[75]:

```
5569
```

In [76]:

```
df16.head()
```

Out[76]:

	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

In [78]:

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

Out[78]:

```
Importados/Indefinidos    19
Bom Jesus                  5
Santa Helena               4
Bonito                     4
Santa Luzia                4
                          ..
Alfenas                    1
Anajás                     1
Colares                    1
Quixabeira                 1
Craíbas                    1
Name: city, Length: 5255, dtype: int64
```

In [79]:

```
df16.head()
```

Out[79]:

	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

In [84]:

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

Out[84]:

```
1880
```

In [85]:

```
df_novos_casos.head()
```

Out[85]:

	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 [86]:

```
# 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[86]:

```
1880
```

In [88]:

```
dfo.head(20)
```

Out[88]:

	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 [90]:

```
list(dfo.place_type.unique())
```

Out[90]:

```
['state', 'city']
```

In [91]:

```
len(dfo)
```

Out[91]:

```
1880
```

In [92]:

```
df3 = dfo[ dfo.place_type == 'city' ].reset_index(drop=True)
len(df3)
```

Out[92]:

```
1857
```

In [94]:

```
1880-1857
```

Out[94]:

```
23
```

In [96]:

```
df3.head(10)
```

Out[96]:

	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 [97]:

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

Out[97]:

	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 [98]:

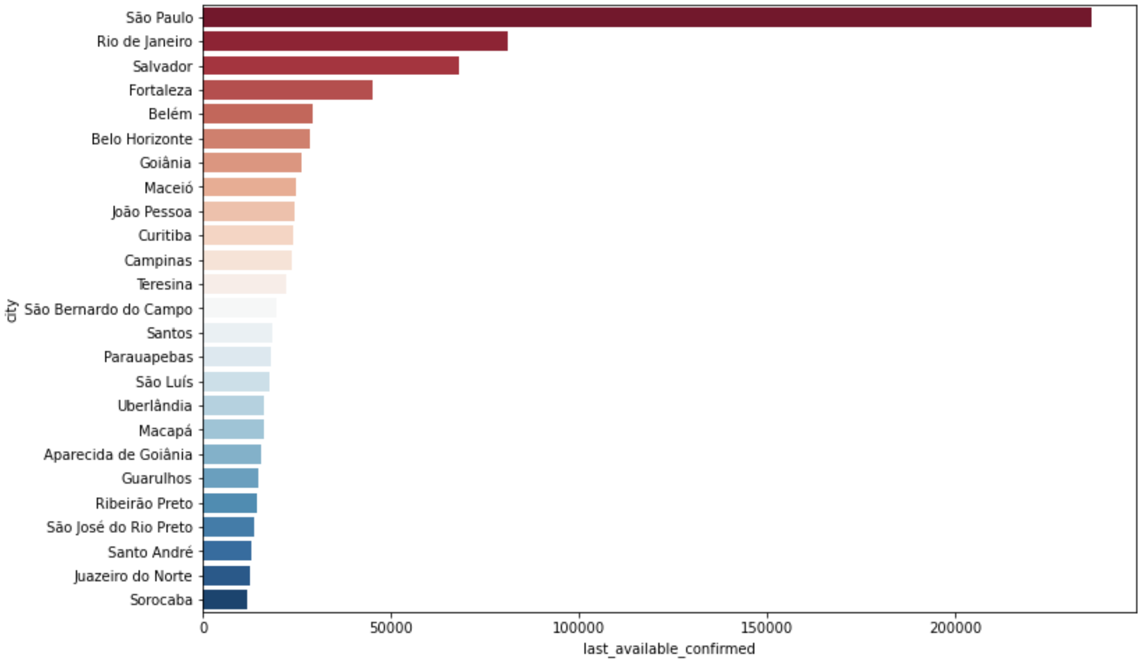
```
dfo[ dfo.city == 'Campina Grande']
```

Out[98]:

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

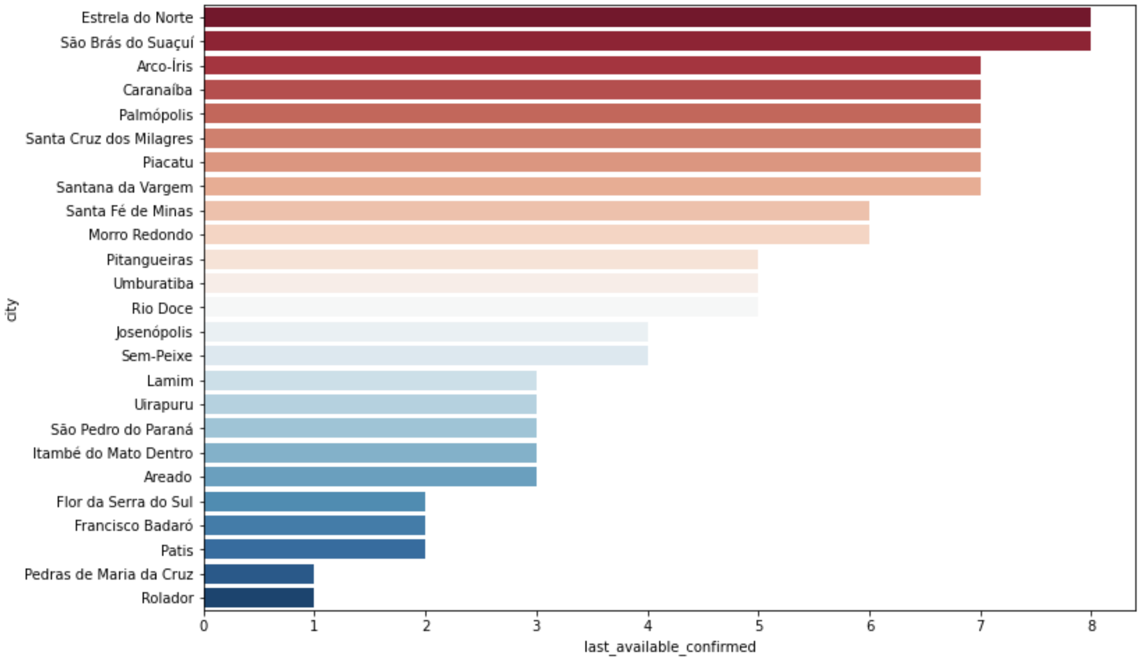
In [100]:

```
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 [101]:

```
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

Dataframes - Continuacao - 25 - Agosto - 2020

In [1]:

```
import pandas as pd
```

In [3]:

```
url2 = 'https://raw.githubusercontent.com/vladimiralencar/Alunos-UEPB-TopicosEsp
eciaisEmBancoDeDados/master/Python-Para-Analise-de-Dados/salarios.csv'

url = "https://github.com/vladimiralencar/Alunos-UEPB-TopicosEspeciaisEmBancoDeD
ados/blob/master/Python-Para-Analise-de-Dados/salarios.csv"
```

In [6]:

```
url = 'https://raw.githubusercontent.com/vladimiralencar/Alunos-UEPB-TopicosEspeciaisEmBancoDeDados/master/Python-Para-Analise-de-Dados/salarios.csv'
url
```

Out[6]:

```
'https://raw.githubusercontent.com/vladimiralencar/Alunos-UEPB-TopicosEspeciaisEmBancoDeDados/master/Python-Para-Analise-de-Dados/salarios.csv'
```

In [7]:

```
df = pd.read_csv(url)
len(df)
```

Out[7]:

```
32182
```

In [8]:

```
df.head()
```

Out[8]:

	Name	Position Title	Department	Employee Annual Salary
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32182 entries, 0 to 32181
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Name                    32181 non-null  object
 1   Position Title          32181 non-null  object
 2   Department              32181 non-null  object
 3   Employee Annual Salary  32181 non-null  object
dtypes: object(4)
memory usage: 1005.8+ KB
```

In [11]:

```
df['salario'] = df['Employee Annual Salary'].apply(lambda x : str(x) )
df.head()
```

Out[11]:

	Name	Position Title	Department	Employee Annual Salary	salario
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00	$88968.00
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00	$80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00	$80778.00
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00	$84780.00
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00	$104736.00

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32182 entries, 0 to 32181
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Name                    32181 non-null  object
 1   Position Title          32181 non-null  object
 2   Department              32181 non-null  object
 3   Employee Annual Salary  32181 non-null  object
 4   salario                 32182 non-null  object
dtypes: object(5)
memory usage: 1.2+ MB
```

In [13]:

```
df['salario'] = df['salario'].apply(lambda x : x.replace('$', '') )
df.head()
```

Out[13]:

	Name	Position Title	Department	Employee Annual Salary	salario
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00	88968.00
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00	80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00	80778.00
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00	84780.00
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00	104736.00

In [16]:

```
df['salario'].max(), df['salario'].min()
```

Out[16]:

```
('nan', '0.96')
```

In [18]:

```
import numpy as np
np.nan # Nulo - Null - Nan - nan
```

Out[18]:

```
nan
```

In [20]:

```
df['salario'] = df['salario'].apply(lambda x : np.nan if x == 'nan' else x )
df.head()
```

Out[20]:

	Name	Position Title	Department	Employee Annual Salary	salario
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00	88968.00
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00	80778.00
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00	80778.00
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00	84780.00
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00	104736.00

In [21]:

```
df['salario'] = df['salario'].apply(lambda x : float(x) )
df.head()
```

Out[21]:

	Name	Position Title	Department	Employee Annual Salary	salario
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00	88968.0
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00	80778.0
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00	80778.0
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00	84780.0
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00	104736.0

In [22]:

```
df.salario.describe()
```

Out[22]:

```
count     32181.000000
mean      79167.525939
std       24462.356678
min           0.960000
25%       69888.000000
50%       83616.000000
75%       91764.000000
max      260004.000000
Name: salario, dtype: float64
```

In [24]:

```
79167*5.60 / 13
```

Out[24]:

```
34102.70769230769
```

In [45]:

```
260004*5.60 / 13
```

Out[45]:

```
112001.72307692307
```

In [28]:

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

In [33]:

```
dfo = df.groupby(['Department']).sum().reset_index()
dfo = dfo.sort_values(by='salario', ascending = False).reset_index(drop=True)
dfo
```

Out[33]:

	Department	salario
0	POLICE	1,106,657,381
1	FIRE	457,971,614
2	WATER MGMNT	155,110,588
3	STREETS & SAN	146,821,191
4	TRANSPORTN	98,344,398
5	AVIATION	96,467,503
6	GENERAL SERVICES	75,486,488
7	OEMC	63,379,954
8	PUBLIC LIBRARY	52,663,130
9	HEALTH	44,719,992
10	FINANCE	39,085,897
11	LAW	30,989,159
12	FAMILY & SUPPORT	26,951,553
13	BUILDINGS	24,127,022
14	CITY COUNCIL	22,367,892
15	COMMUNITY DEVELOPMENT	17,754,150
16	BUSINESS AFFAIRS	12,889,272
17	DoIT	10,136,460
18	MAYOR'S OFFICE	8,120,517
19	IPRA	7,358,004
20	PROCUREMENT	6,303,888
21	BOARD OF ELECTION	6,252,156
22	CULTURAL AFFAIRS	6,091,348
23	CITY CLERK	5,439,045
24	HUMAN RESOURCES	4,889,112
25	INSPECTOR GEN	4,416,264
26	BUDGET & MGMT	3,776,940
27	ANIMAL CONTRL	3,699,370
28	ADMIN HEARNG	2,898,456
29	DISABILITIES	2,109,660
30	TREASURER	1,919,517
31	HUMAN RELATIONS	1,513,356
32	BOARD OF ETHICS	750,852
33	POLICE BOARD	158,136
34	LICENSE APPL COMM	69,888

In [34]:

```
dfo.head()
```

Out[34]:

	Department	salario
0	POLICE	1,106,657,381
1	FIRE	457,971,614
2	WATER MGMNT	155,110,588
3	STREETS & SAN	146,821,191
4	TRANSPORTN	98,344,398

In [38]:

```
del dfo['salario_mensal_R$']
dfo['salario_mensal_total_R$'] = (dfo['salario'] / 13) * 5.60
dfo
```

```
Out[38]:
```

	Department	salario	salario_mensal_total_R$
0	POLICE	1,106,657,381	476,713,949
1	FIRE	457,971,614	197,280,080
2	WATER MGMNT	155,110,588	66,816,869
3	STREETS & SAN	146,821,191	63,246,052
4	TRANSPORTN	98,344,398	42,363,740
5	AVIATION	96,467,503	41,555,232
6	GENERAL SERVICES	75,486,488	32,517,257
7	OEMC	63,379,954	27,302,134
8	PUBLIC LIBRARY	52,663,130	22,685,656
9	HEALTH	44,719,992	19,263,997
10	FINANCE	39,085,897	16,837,002
11	LAW	30,989,159	13,349,176
12	FAMILY & SUPPORT	26,951,553	11,609,900
13	BUILDINGS	24,127,022	10,393,179
14	CITY COUNCIL	22,367,892	9,635,400
15	COMMUNITY DEVELOPMENT	17,754,150	7,647,941
16	BUSINESS AFFAIRS	12,889,272	5,552,302
17	DoIT	10,136,460	4,366,475
18	MAYOR'S OFFICE	8,120,517	3,498,069
19	IPRA	7,358,004	3,169,602
20	PROCUREMENT	6,303,888	2,715,521
21	BOARD OF ELECTION	6,252,156	2,693,236
22	CULTURAL AFFAIRS	6,091,348	2,623,965
23	CITY CLERK	5,439,045	2,342,973
24	HUMAN RESOURCES	4,889,112	2,106,079
25	INSPECTOR GEN	4,416,264	1,902,391
26	BUDGET & MGMT	3,776,940	1,626,990
27	ANIMAL CONTRL	3,699,370	1,593,575
28	ADMIN HEARNG	2,898,456	1,248,566
29	DISABILITIES	2,109,660	908,777
30	TREASURER	1,919,517	826,869
31	HUMAN RELATIONS	1,513,356	651,907
32	BOARD OF ETHICS	750,852	323,444
33	POLICE BOARD	158,136	68,120
34	LICENSE APPL COMM	69,888	30,106

In [39]:

```
df.head()
```

Out[39]:

	Name	Position Title	Department	Employee Annual Salary	salario
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00	88,968
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00	80,778
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00	80,778
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00	84,780
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00	104,736

In [40]:

```
dfo = df.groupby(['Position Title']).mean().reset_index()
dfo = dfo.sort_values(by='salario', ascending = False).reset_index(drop=True)
dfo
```

Out[40]:

	Position Title	salario
0	SUPERINTENDENT OF POLICE	260,004
1	MAYOR	216,210
2	FIRE COMMISSIONER	202,728
3	FIRST DEPUTY FIRE COMMISSIONER	197,736
4	FIRST DEPUTY SUPERINTENDENT	197,736
...	...	...
1089	HOSPITALITY WORKER	9,516
1090	PROGRAM AIDE	9,360
1091	TITLE V PROGRAM TRAINEE I	8,580
1092	SENIOR COMPANION	2,756
1093	FOSTER GRANDPARENT	2,756

1094 rows × 2 columns

In [43]:

```
dfo['salario_mensal_total_R$'] = (dfo['salario'] / 13) * 5.60
dfo.head(50)
```

Out[43]:

	Position Title	salario	salario_mensal_total_R$
0	SUPERINTENDENT OF POLICE	260,004	112,002
1	MAYOR	216,210	93,137
2	FIRE COMMISSIONER	202,728	87,329
3	FIRST DEPUTY FIRE COMMISSIONER	197,736	85,179
4	FIRST DEPUTY SUPERINTENDENT	197,736	85,179
5	DEPUTY FIRE COMMISSIONER	187,680	80,847
6	CHIEF	186,846	80,488
7	ASST DEPUTY FIRE COMMISSIONER	185,352	79,844
8	COMMISSIONER OF HEALTH	177,000	76,246
9	CHIEF OF STAFF	174,996	75,383
10	PSYCHIATRIST	174,720	75,264
11	CORPORATION COUNSEL	173,664	74,809
12	CHIEF FINANCIAL OFFICER	169,992	73,227
13	BUDGET DIR	169,992	73,227
14	COMMISSIONER OF WATER MGMT	169,512	73,021
15	COMMISSIONER OF TRANSPORTATION	169,500	73,015
16	DIR OF INTERGOVERNMENTAL AFFAIRS	168,996	72,798
17	DEPUTY CHIEF	168,906	72,760
18	EXEC DIR EMERG MGMT & COMM	167,796	72,281
19	CHIEF PROCUREMENT OFFICER	167,220	72,033
20	COMMISSIONER OF CHICAGO PUBLIC LIBRARY	167,004	71,940
21	CITY COMPTROLLER	165,000	71,077
22	PRESS SECRETARY	162,492	69,997
23	SUPERINTENDENT'S CHIEF OF STAFF	162,012	69,790
24	INSPECTOR GENERAL	161,856	69,723
25	CHIEF ADMINISTRATOR	161,856	69,723
26	FIRST DEPUTY CHIEF OF STAFF	159,996	68,921
27	DISTRICT CHIEF	158,308	68,194
28	COMMANDER	157,789	67,971
29	COORD OF FIRE AWARENESS	157,776	67,965
30	COORD OF SPECIAL EVENTS LIAISON	157,776	67,965
31	COMMANDING FIRE MARSHAL	157,776	67,965
32	DEPUTY CHIEF OF EMPLOYEE RELATIONS	157,776	67,965
33	COORD OF AIR MASK SERVICES	157,776	67,965
34	COMMISSIONER OF STREETS AND SANITATION	157,092	67,670
35	COMMISSIONER OF FAMILY AND SUPPORT SERVICES	157,092	67,670
36	COMMISSIONER OF FLEET & FACILITY MANAGEMENT	157,092	67,670

	Position Title	salario	salario_mensal_total_R$
37	COMMISSIONER OF BUILDINGS	157,092	67,670
38	COMMISSIONER OF BUS AFAIRS AND CONSUMER PROT	157,092	67,670
39	COMMISSIONER OF HOUSING & ECONOMIC DEV	156,504	67,417
40	DIR OF ADMINISTRATIVE HEARINGS	156,420	67,381
41	ASST DEPUTY CHIEF PARAMEDIC	156,360	67,355
42	DEPUTY DISTRICT CHIEF	156,360	67,355
43	COMMISSIONER OF CULTURAL AFFAIRS/SPEC EVENTS	155,040	66,786
44	CHIEF INFORMATION OFFICER	154,992	66,766
45	EXECUTIVE DIR	154,992	66,766
46	GENERAL COUNSEL	154,242	66,443
47	COMMISSIONER OF HUMAN RESOURCES	151,572	65,293
48	DIR OF HUMAN RESOURCES	150,396	64,786
49	CHIEF ADMINISTRATIVE OFFICER	149,514	64,406

In [44]:

```
dfo.tail(25)
```

Out[44]:

	Position Title	salario	salario_mensal_total_R$
1069	CLERK CITY COUNCIL	24,939	10,743
1070	LIBRARY ASSOCIATE - HOURLY	24,601	10,597
1071	LAW CLERK	24,361	10,494
1072	READER	23,332	10,051
1073	ANIMAL CARE CLERK - HOURLY	22,641	9,753
1074	ALDERMANIC AIDE	22,337	9,622
1075	AVIATION SECURITY OFFICER - HOURLY	22,121	9,529
1076	CRIMES SURVEILLANCE SPECIALIST	19,677	8,476
1077	TRAFFIC CONTROL AIDE-HOURLY	19,654	8,466
1078	CROSSING GUARD	19,029	8,197
1079	ELDERLY AIDE III HOURLY	18,803	8,100
1080	SENIOR LIBRARY CLERK - HOURLY	17,545	7,558
1081	SERVICE COORD AIDE	17,139	7,383
1082	LIBRARY CLERK - HOURLY	15,995	6,890
1083	STUDENT INTERN	15,933	6,863
1084	CROSSING GUARD - PER AGREEMENT	15,731	6,776
1085	CUSTODIAL WORKER - PART TIME	13,468	5,802
1086	LIBRARY PAGE	12,359	5,324
1087	STUDENT INTERN - CITY CLERK	10,920	4,704
1088	POLICE CADET	9,630	4,148
1089	HOSPITALITY WORKER	9,516	4,099
1090	PROGRAM AIDE	9,360	4,032
1091	TITLE V PROGRAM TRAINEE I	8,580	3,696
1092	SENIOR COMPANION	2,756	1,187
1093	FOSTER GRANDPARENT	2,756	1,187

In [53]:

```
dfp = df['Position Title'].value_counts().reset_index()
dfp.head(50)
```

Out[53]:

	index	Position Title
0	POLICE OFFICER	9489
1	FIREFIGHTER-EMT	1191
2	SERGEANT	1138
3	FIREFIGHTER	970
4	POLICE OFFICER (ASSIGNED AS DETECTIVE)	808
5	MOTOR TRUCK DRIVER	743
6	SANITATION LABORER	730
7	POOL MOTOR TRUCK DRIVER	631
8	CROSSING GUARD	594
9	CONSTRUCTION LABORER	400
10	TRAFFIC CONTROL AIDE-HOURLY	383
11	PARAMEDIC	358
12	LIEUTENANT-EMT	343
13	LIEUTENANT	338
14	FIRE ENGINEER-EMT	283
15	FIREFIGHTER/PARAMEDIC	269
16	PARAMEDIC I/C	247
17	AVIATION SECURITY OFFICER	235
18	HOISTING ENGINEER	223
19	POLICE COMMUNICATIONS OPERATOR II	221
20	ELECTRICAL MECHANIC	210
21	DETENTION AIDE	206
22	GENERAL LABORER - DSS	169
23	OPERATING ENGINEER-GROUP A	165
24	MACHINIST (AUTOMOTIVE)	158
25	OPERATING ENGINEER-GROUP C	157
26	SENIOR DATA ENTRY OPERATOR	156
27	FIRE ENGINEER	155
28	CONCRETE LABORER	154
29	CAPTAIN-EMT	145
30	LABORER	140
31	POLICE COMMUNICATIONS OPERATOR I	140
32	FIREFIGHTER (PER ARBITRATORS AWARD)-PARAMEDIC	139
33	CROSSING GUARD - PER AGREEMENT	138
34	ADMINISTRATIVE ASST II	136
35	STAFF ASST	136
36	FOSTER GRANDPARENT	134

	index	Position Title
37	ASST CORPORATION COUNSEL	129
38	LIBRARY PAGE	129
39	LIBRARIAN I	124
40	PLUMBER	118
41	STAFF ASST TO THE ALDERMAN	114
42	ADMINISTRATIVE ASST III	114
43	PARKING ENFORCEMENT AIDE	112
44	CUSTODIAL WORKER	110
45	POLICE OFFICER / FLD TRNG OFFICER	108
46	ASPHALT LABORER	107
47	CLERK III	100
48	LABORER - APPRENTICE	98
49	LEGISLATIVE AIDE	95

In [59]:

```
len(list(df['Position Title'].unique()))
```

Out[59]:

```
1095
```

In [54]:

```
df.head()
```

Out[54]:

	Name	Position Title	Department	Employee Annual Salary	salario
0	AARON, ELVIA J	WATER RATE TAKER	WATER MGMNT	$88968.00	88,968
1	AARON, JEFFERY M	POLICE OFFICER	POLICE	$80778.00	80,778
2	AARON, KARINA	POLICE OFFICER	POLICE	$80778.00	80,778
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	GENERAL SERVICES	$84780.00	84,780
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	$104736.00	104,736

In [56]:

```
df.salario.sum() / 13 * 5.6
```

Out[56]:

```
1097466527.1187692
```

In [ ]: