_sprint #4_v.0.1_fabio_goncalves

August 16, 2021

1 Fábio Gonçalves

2 Sprint #4 - Covid API Data

https://covid19-api.vost.pt/

HTML_CODES
https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status

Usage of * resquests * pandas * graphics * functions * widgets

Objectives 1. From COVID API import data using funcions and requests 1. Convert `['confirmados_novos', 'internados', 'internados_uci']` to pandas 1. Describe `['confirmados_novos', 'internados', 'internados_uci']` 1. Make a Box plot of "['confirmados_novos', 'internados', 'internados_uci']"' 1. Make a evolution plot of confirmados_novos with moving average 1. Make a Widget version with selection dates and combo box 1. Make a summary conclusion

2.0.1 from COVID API import data using funcions and requests

2.0.2 Convert ['confirmados_novos', 'internados', 'internados_uci'] to pandas

[1]:
```
import requests
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib as mpl
from datetime import datetime
import ipywidgets as widgets
from ipywidgets import interact_manual, IntSlider, Select

#imputs do utilizador
data_inicio=input('Introduza a data inicial a pesquisar(formato dd-mm-yyyy?')
data_fim=input('Introduza a data final a pesquisar(formato dd-mm-yyyy?')

#import do data api
result = requests.get(f'https://covid19-api.vost.pt/Requests/get_entry/
 ↪{data_inicio}_until_{data_fim}')

#verificar se import foi feito corretamente
```

```
if result.status_code==200:
    covdid=result.json()

else:
    print('Error fetching data')

#conversão do dados num dataframe
data_pack = ['confirmados_novos', 'internados', 'internados_uci']
datas = result.json()['data'].values()
dict_ = {}
for el in data_pack:
    dict_[el] =result.json()[(el)].values()
    df = pd.DataFrame(dict_ , datas)
    df.index=pd.to_datetime(df.index,format='%d-%m-%Y')
    df['avg_confirmados_novos'] = round(df['confirmados_novos'].
 ↪rolling(window=3).mean(),2)

#Print tabela de casos covid entre datas
df
```

```
Introduza a data inicial a pesquisar(formato dd-mm-yyyy? 15-03-2020
Introduza a data final a pesquisar(formato dd-mm-yyyy? 15-08-2021
```

[1]:

	confirmados_novos	internados	internados_uci	\
2020-03-15	76	139.0	9.0	
2020-03-16	86	139.0	18.0	
2020-03-17	117	206.0	17.0	
2020-03-18	194	89.0	20.0	
2020-03-19	143	89.0	20.0	
…	…	…	…	
2021-08-10	2232	829.0	186.0	
2021-08-11	2948	785.0	181.0	
2021-08-12	2708	754.0	169.0	
2021-08-13	2598	732.0	162.0	
2021-08-14	2571	727.0	161.0	

	avg_confirmados_novos
2020-03-15	NaN
2020-03-16	NaN
2020-03-17	93.00
2020-03-18	132.33
2020-03-19	151.33
…	…
2021-08-10	1769.33
2021-08-11	2091.33
2021-08-12	2629.33
2021-08-13	2751.33

```
2021-08-14                2625.67

[518 rows x 4 columns]
```

2.0.3 Describe [‘confirmados_novos’, ‘internados’, ‘internados_uci’]

[2]:
```
# imprimo o Describe
round(df.describe(),2)
```

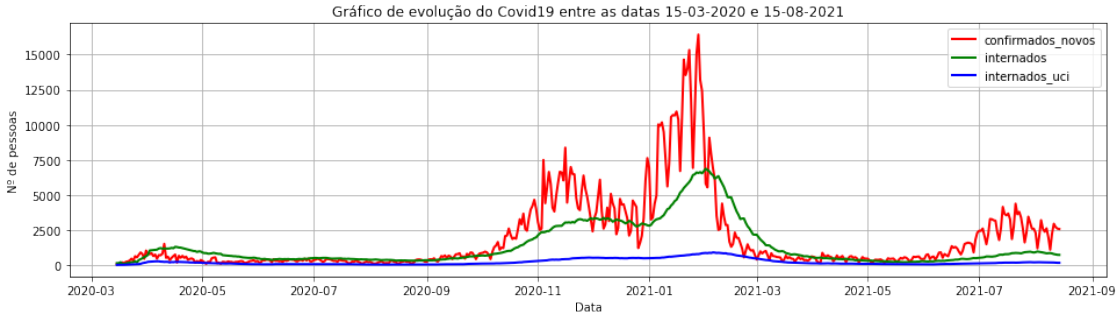
[2]:

	confirmados_novos	internados	internados_uci	avg_confirmados_novos
count	518.00	518.00	518.00	516.00
mean	1932.33	1365.39	222.65	1934.67
std	2740.93	1540.63	217.57	2676.32
min	76.00	89.00	9.00	93.00
25%	344.25	415.25	70.00	348.50
50%	640.00	660.00	124.50	630.33
75%	2588.25	1649.75	283.75	2720.58
max	16432.00	6869.00	904.00	14901.67

[3]:
```
#plot dos resultados acima

plt.figure(figsize=(16,4))
plt.plot(df['confirmados_novos'],'r',linewidth=2, label='confirmados_novos')
plt.plot(df['internados'],'g',linewidth=2, label='internados')
plt.plot(df['internados_uci'],'b',linewidth=2, label='internados_uci')
plt.grid()
plt.title(f'Gráfico de evolução do Covid19 entre as datas {data_inicio} e␣
 ↪{data_fim}')
plt.ylabel('Nº de pessoas')
plt.xlabel('Data')
plt.legend()
```

[3]:
```
<matplotlib.legend.Legend at 0x7f21ede10e50>
```

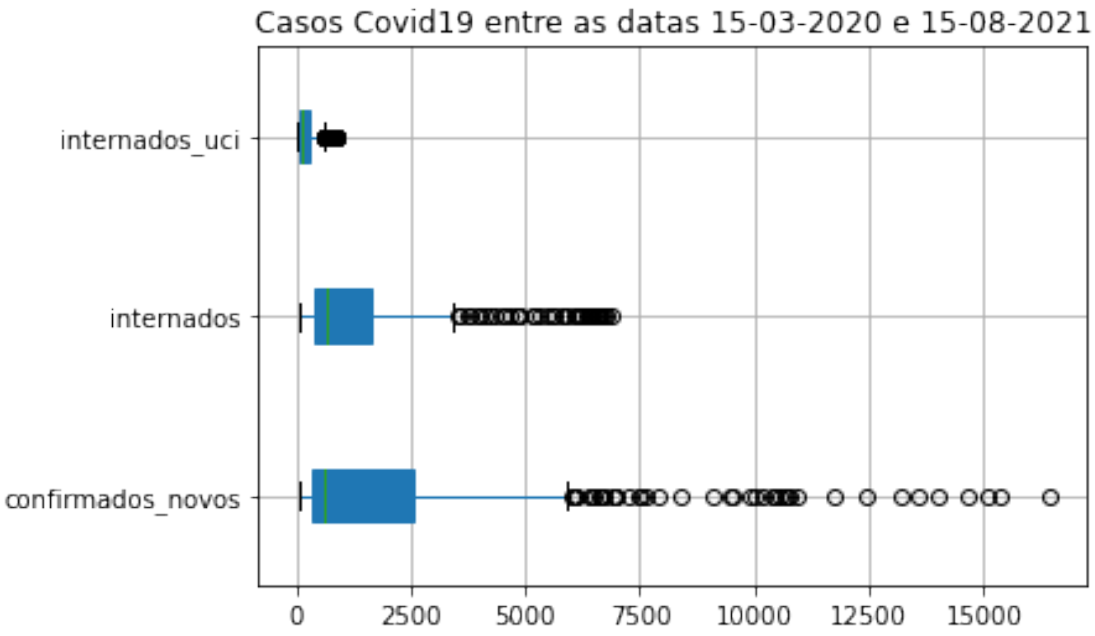


2.0.4 Make a Box plot of [‘confirmados_novos’, ‘internados’, ‘internados_uci’]

[4]:
```
#boxplot dos resultdos acima

df.boxplot(data_pack, patch_artist=True,vert=False)
plt.title(f'Casos Covid19 entre as datas {data_inicio} e {data_fim}')
```

[4]: Text(0.5, 1.0, 'Casos Covid19 entre as datas 15-03-2020 e 15-08-2021')

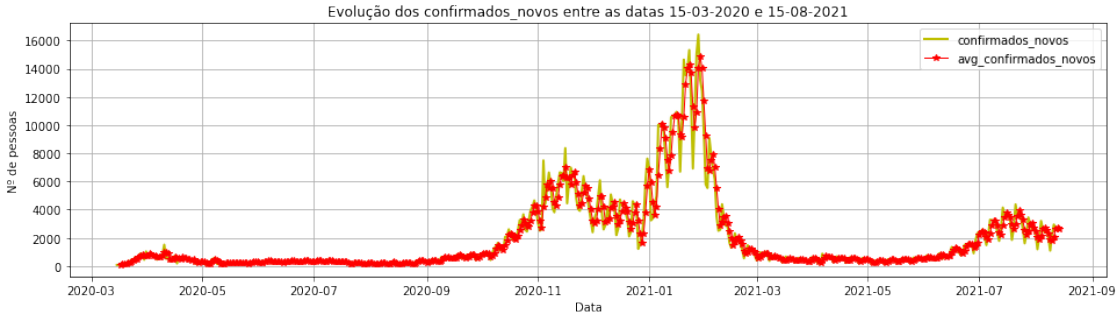


2.0.5 Make a evolution plot of confirmados_novos with moving average

[5]:
```
#plot com a média deslizante dos Confirmados_novos

plt.figure(figsize=(16,4))
plt.plot(df['confirmados_novos'],'y',linewidth=2, label='confirmados_novos')
plt.plot(df['avg_confirmados_novos'],'r-*',linewidth=1,␣
 ↪label='avg_confirmados_novos')
plt.grid()
plt.title(f'Evolução dos confirmados_novos entre as datas {data_inicio} e␣
 ↪{data_fim}')
plt.ylabel('Nº de pessoas')
plt.xlabel('Data')
plt.legend()
```

[5]: <matplotlib.legend.Legend at 0x7f21e59efe50>



2.0.6 Make a Widget version with selection dates and combo box

[6]:
```
# função com grafico interativo com combobox

def grafico_widgets(pack):
    data_pack = ['confirmados_novos', 'internados', 'internados_uci']
    data_pack=pack
    plt.figure(figsize=(16,4))
    plt.plot(df[pack],'b--',linewidth=2, label=pack)
    plt.grid()
    plt.title(f'Gráfico de evolução do Covid19 entre as datas {data_inicio} e␣
 ↪{data_fim}')
    plt.ylabel('Nº de pessoas')
    plt.xlabel('Data')
    plt.legend()
    plt.show()

interact_manual(grafico_widgets,
                pack=Select(options=['confirmados_novos', 'internados',␣
 ↪'internados_uci', 'avg_confirmados_novos']))
```

```
interactive(children=(Select(description='pack', options=('confirmados_novos',␣
 ↪'internados', 'internados_uci',…
```

[6]:
```
<function __main__.grafico_widgets(pack)>
```

2.0.7 Make a summary conclusion

[7]:
```
def conclusion(pack):
    data_pack = ['confirmados_novos', 'internados', 'internados_uci']
    data_pack=pack
    max_ =int(df[data_pack].max())
    max_dt= df[data_pack].idxmax()
    min_dt = df[data_pack].idxmin()
    min_ = int(df[data_pack].min())
```

```
    med =round(df[data_pack].mean(),2)

    print (f'Entre as datas {data_inicio} e {data_fim} o dia com mais␣
↪{data_pack} foi no dia {datetime.strftime(max_dt,"%Y-%m-%d")} com {max_}␣
↪casos.')
    print (f'Entre as datas {data_inicio} e {data_fim} o dia com menos␣
↪{data_pack} foi no dia {datetime.strftime(min_dt,"%Y-%m-%d")} com {min_}␣
↪casos.')
    print (f'Entre as datas {data_inicio} e {data_fim} a  média de {data_pack}␣
↪foi de {med} casos')

interact_manual(conclusion,
                pack=Select(options=['confirmados_novos', 'internados',␣
↪'internados_uci']))
```

```
interactive(children=(Select(description='pack', options=('confirmados_novos',␣
↪'internados', 'internados_uci')…
```

[7]:
```
<function __main__.conclusion(pack)>
```

[8]:
```
#
```