Analisi COVID-19 - Federico

March 18, 2020

1 Analisi Covid_19

Analisi a livello regionale:

- Ogni regione grafico singolo
- Todo: aggiungere mortalità per regione ##### Analisi a livello nazionale
- Casi totali Italia
- Aggiungere proporzione guariti/terapia intensiva

TODO:

- Generazione giornalieria automatica
- Devo aggiustare il codice che è disordinato e senza commenti
- Scrivere un Readme.md come si deve

[1]:
```
# Importo librerie e apro primo .csv (livello regionale)

import pandas as pd
import matplotlib.pyplot as plt
import pylab
df = pd.read_csv('COVID-19/dati-regioni/dpc-covid19-ita-regioni.csv')
```

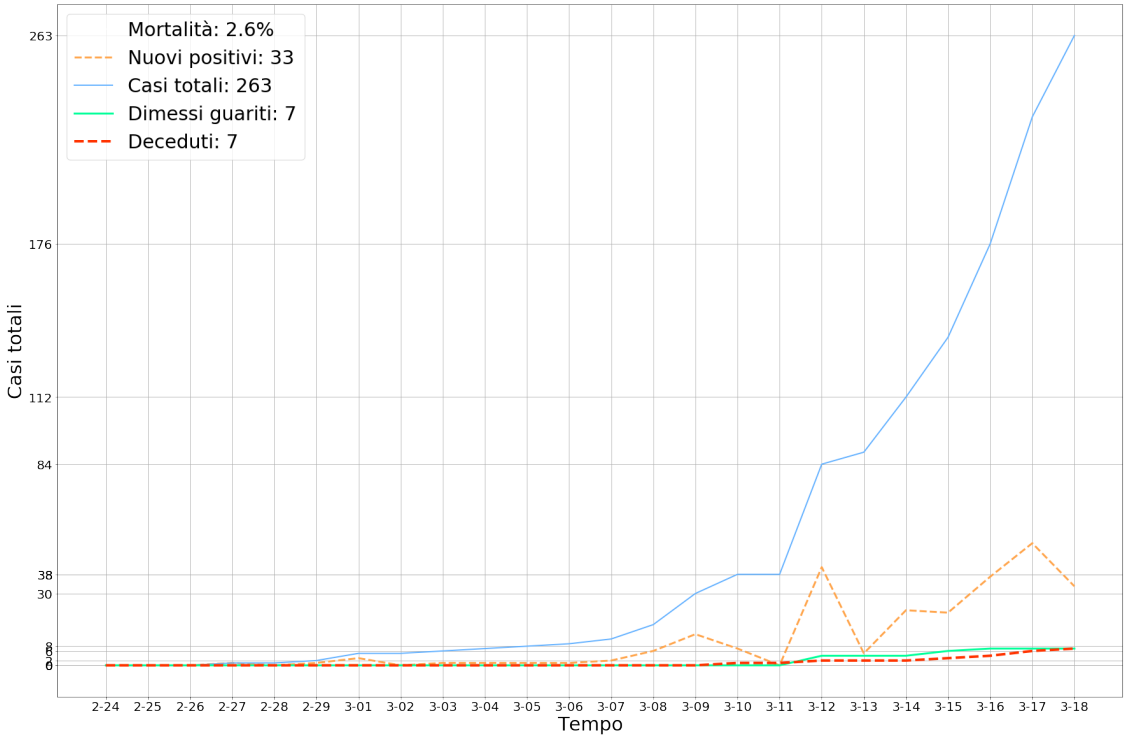
[2]:
```
# Lista regioni

regione_tot = ['Abruzzo', 'Basilicata', 'P.A. Bolzano', 'Calabria', 'Campania',␣
 ↪'Emilia Romagna',
               'Friuli Venezia Giulia', 'Lazio', 'Liguria', 'Lombardia',␣
 ↪'Marche', 'Molise', 'Piemonte', 'Puglia',
               'Sardegna', 'Sicilia', 'Toscana', 'P.A. Trento', 'Umbria',␣
 ↪"Valle d'Aosta", 'Veneto'
]
df.columns
```

[2]:
```
Index(['data', 'stato', 'codice_regione', 'denominazione_regione', 'lat',
       'long', 'ricoverati_con_sintomi', 'terapia_intensiva',
       'totale_ospedalizzati', 'isolamento_domiciliare',
       'totale_attualmente_positivi', 'nuovi_attualmente_positivi',
       'dimessi_guariti', 'deceduti', 'totale_casi', 'tamponi'],
      dtype='object')
```

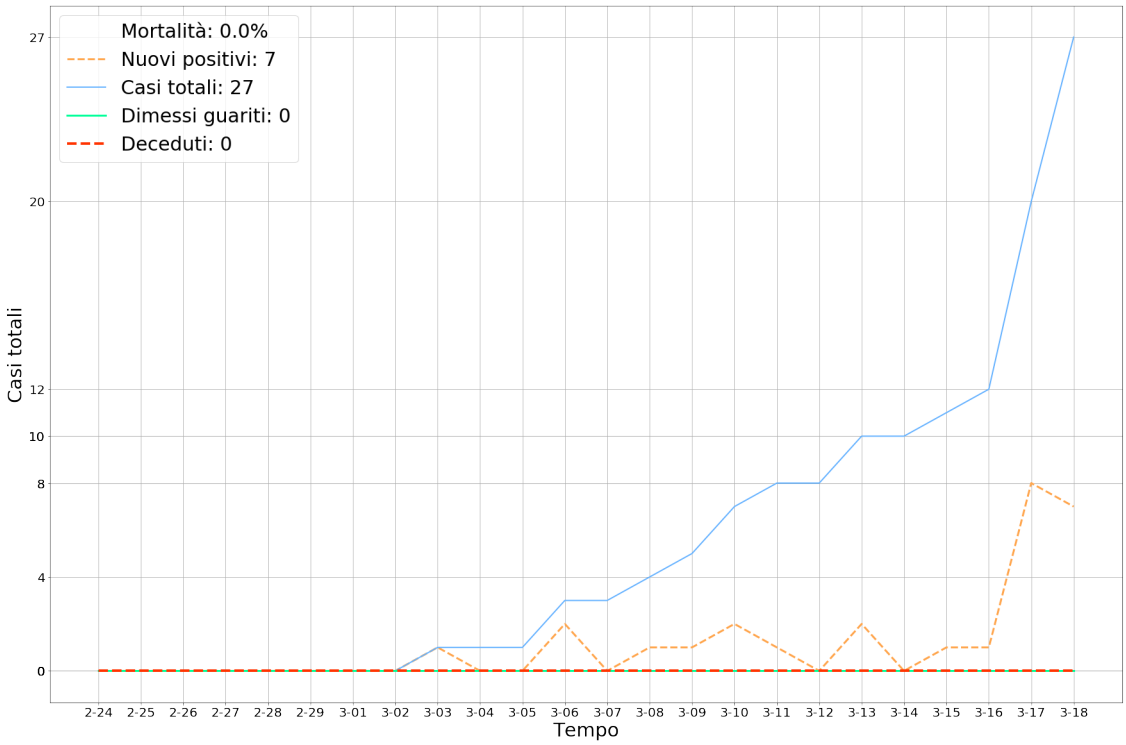
[3]:
```
# Manipolazione lista regioni per ottenere i dati raggruppati per regione.

for z in regione_tot:
    regione = df.loc[df['denominazione_regione'] == z]
    x1 = regione.data
    x2 = regione.totale_casi
    x3 = regione.terapia_intensiva
    x4 = regione.deceduti
    x5 = regione.dimessi_guariti
    x6 = regione.nuovi_attualmente_positivi
    ticks = []
    ticks_1 = []

    x = []
    for f in x1:
        x.append(f[6:10])

    legenda_casi_totali = []
    for casi in x2:
        legenda_casi_totali.append(casi)

    for w in legenda_casi_totali:
        if w % 2 == 0:
            ticks.append(w)
        else:
            pass


    legenda_terapia_intensiva = []
    for casi in x3:
        legenda_terapia_intensiva.append(casi)

    legenda_deceduti = []
    for casi in x4:
        legenda_deceduti.append(casi)

    legenda_guariti = []
    for casi in x5:
        legenda_guariti.append(casi)

    legenda_nuovi_positivi = []
    for casi in x6:
        legenda_nuovi_positivi.append(casi)

    ticks_1.append(legenda_casi_totali[-1])
    ticks.extend(ticks_1)
```

```
    totale_casi = regione.totale_casi
    terapia_intensiva = regione.terapia_intensiva
    deceduti = regione.deceduti
    dimessi_guariti = regione.dimessi_guariti
    nuovi_positivi = regione.nuovi_attualmente_positivi

    plt.rcParams["figure.figsize"]=30,20

    plt.rc('ytick', labelsize=20)
    plt.rc('xtick', labelsize=20)
    plt.rc('axes', labelsize=30)

    plt.title("{}".format(z), fontsize=100)
    plt.yticks(ticks)

    plt.xlabel("Tempo")
    plt.ylabel("Casi totali")

    toll_1_tot = int(legenda_casi_totali[-1])
    toll_2_tot = int(legenda_deceduti[-1])
    death_toll = (toll_2_tot/toll_1_tot)*100
    conv_deth_toll = str(death_toll)

    plt.plot(death_toll, color='#FFFFFF', label="Mortalità: {}%".
↪format(conv_deth_toll[:3]))
    plt.plot(x, nuovi_positivi, color="#ffa64d", linewidth=3, linestyle="--",␣
↪label="Nuovi positivi: {}".format(legenda_nuovi_positivi[-1]))
    plt.plot(x,totale_casi, color='#66b3ff', linewidth=2, label='Casi totali:␣
↪{}'.format(legenda_casi_totali[-1]))
    #plt.plot(x,terapia_intensiva, color='#ff9900', linewidth=2, label='Terapia␣
↪Intensiva: {}'.format(legenda_terapia_intensiva[-1]))
    plt.plot(x,dimessi_guariti, color='#00ff99', linewidth=3, label='Dimessi␣
↪guariti: {}'.format(legenda_guariti[-1]))
    plt.plot(x,deceduti, color='#ff3300', linestyle="--", linewidth=4,␣
↪label='Deceduti: {}'.format(legenda_deceduti[-1]))
    plt.legend(prop={'size': 30})
    plt.grid()

    # Togliendo il commento tutti i grafici verranno salvati in formato .png in␣
↪locale
    # plt.savefig('Estrazioni_reg/{}.png'.format(z))
    plt.show()
    plt.clf()
```

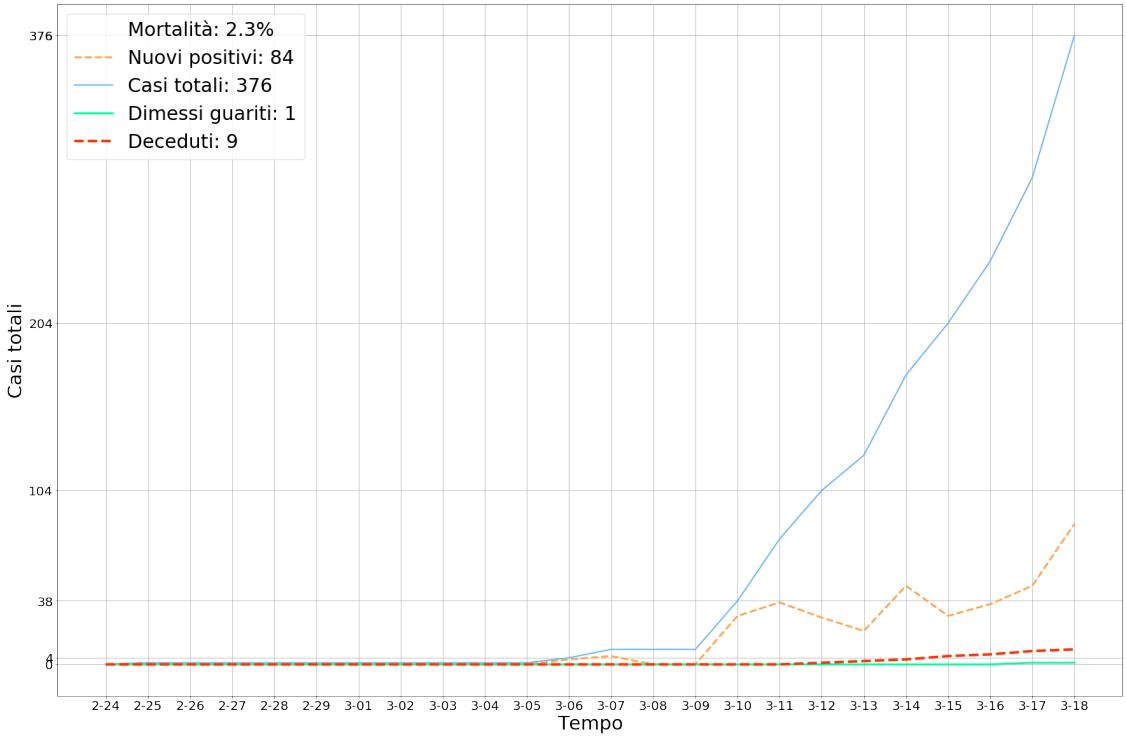
Abruzzo



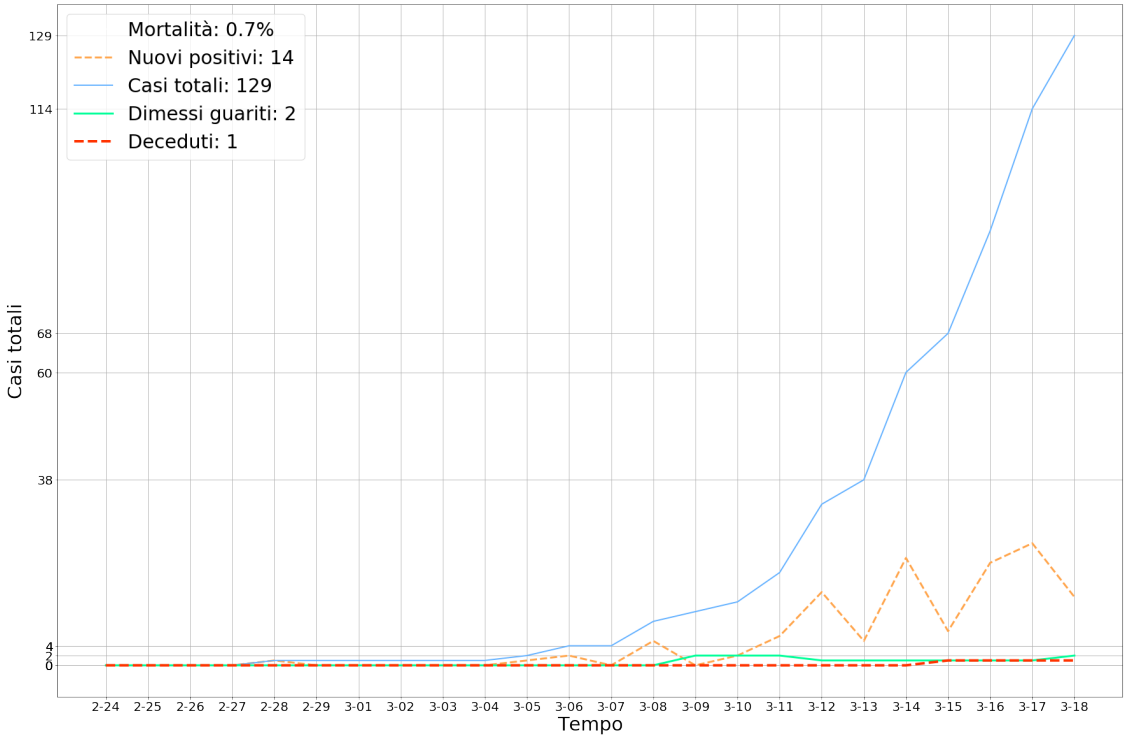
Basilicata



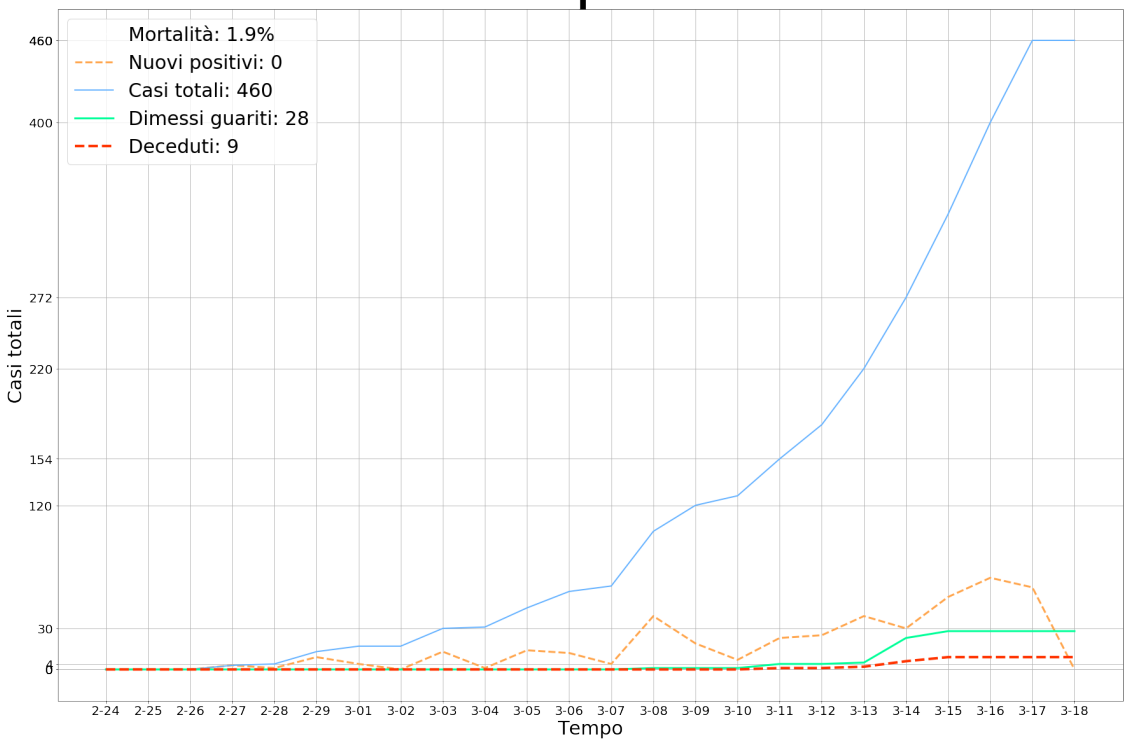
P.A. Bolzano


Mortalità: 2.3%
Nuovi positivi: 84
Casi totali: 376
Dimessi guariti: 1
Deceduti: 9
Casi totali
376
204
104
38
4
0
2-24
2-25
2-26
2-27
2-28
2-29
3-01
3-02
3-03
3-04
3-05
3-06
3-07
3-08
3-09
3-10
3-11
3-12
3-13
3-14
3-15
3-16
3-17
3-18
Tempo

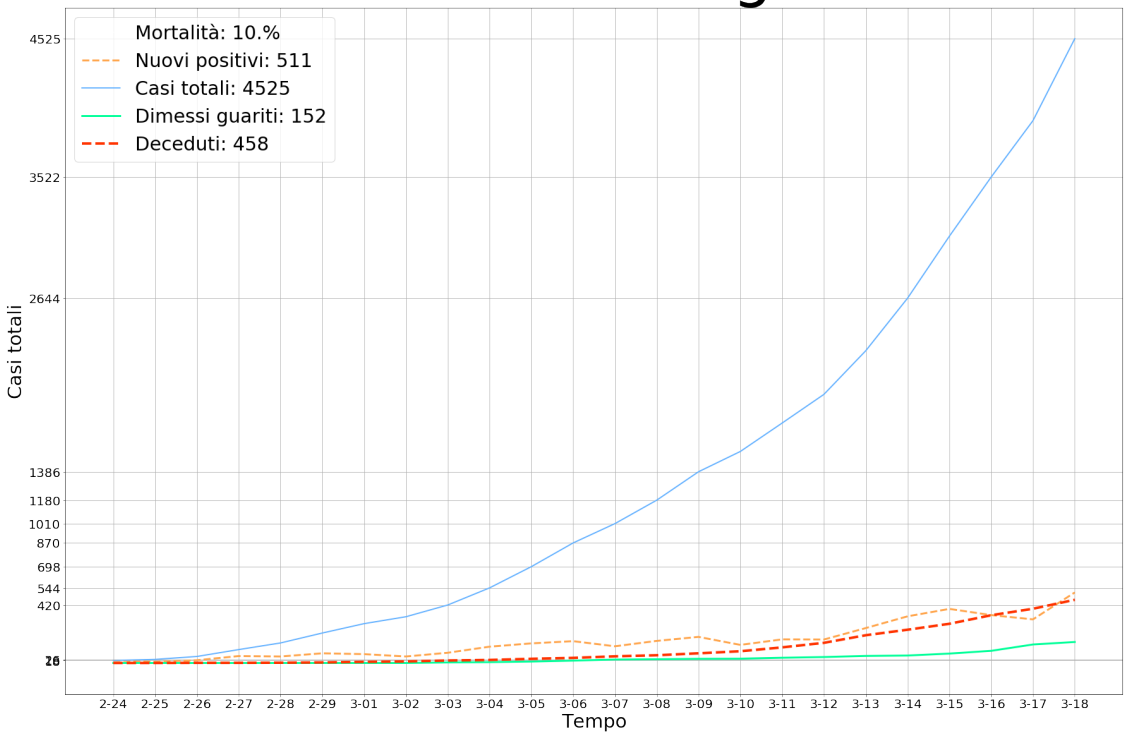
Calabria



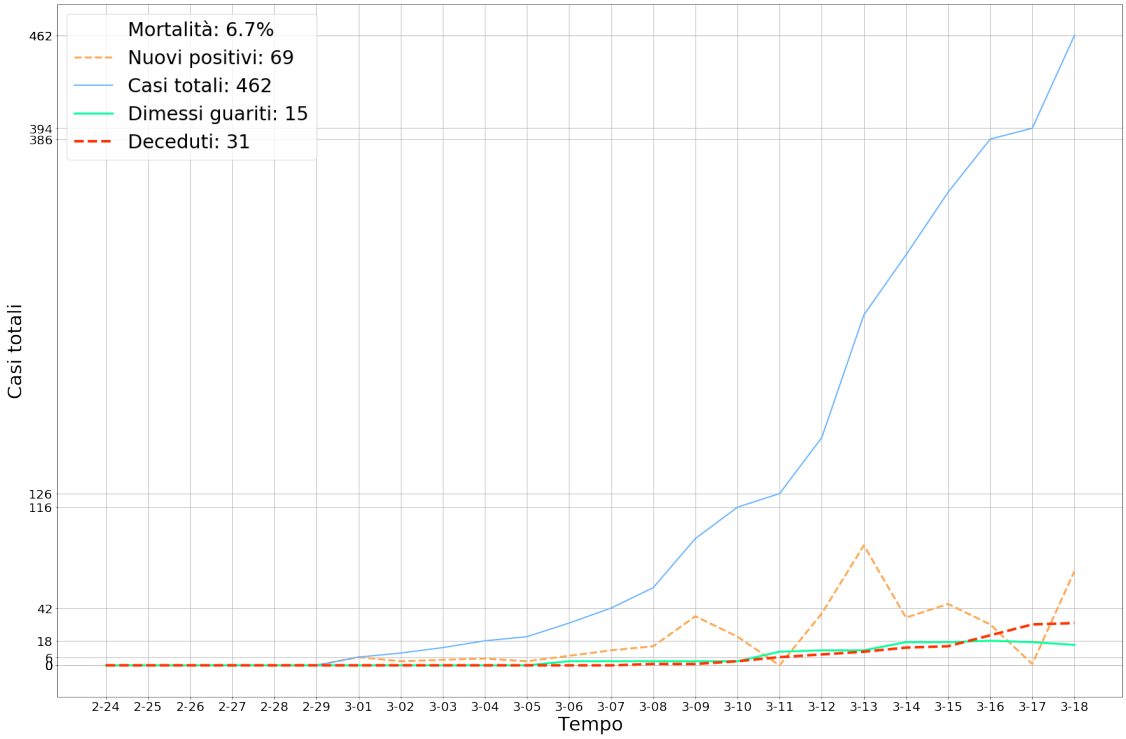
Campania



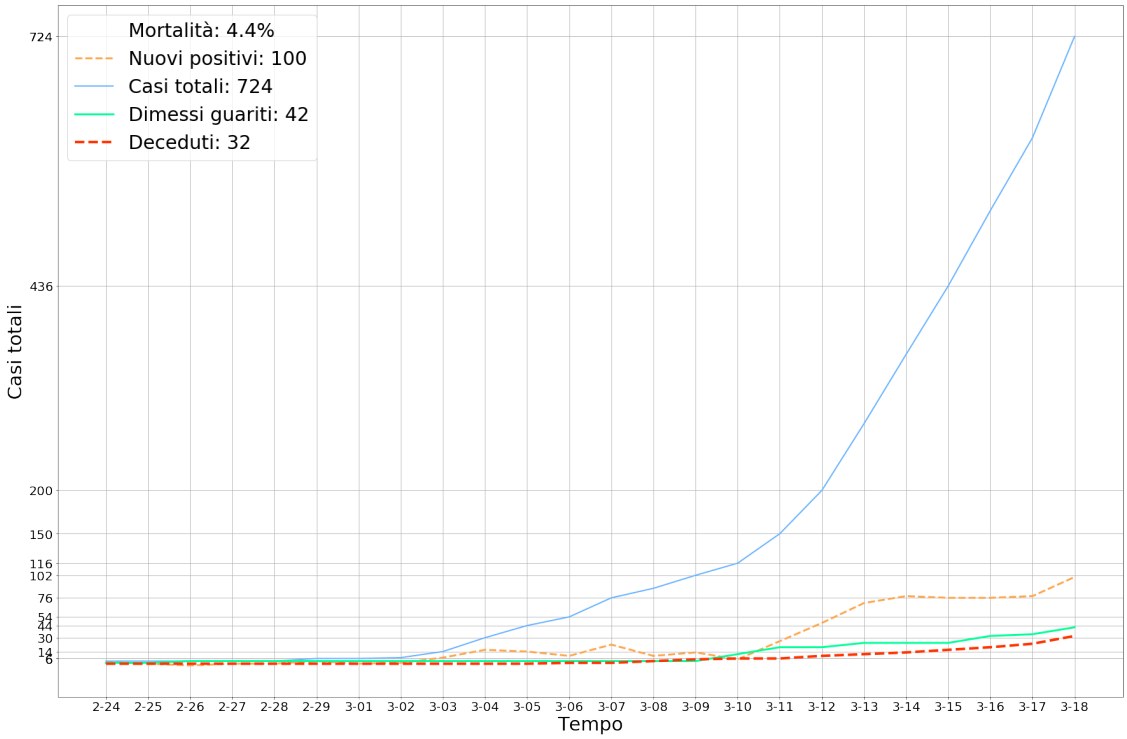
Emilia Romagna



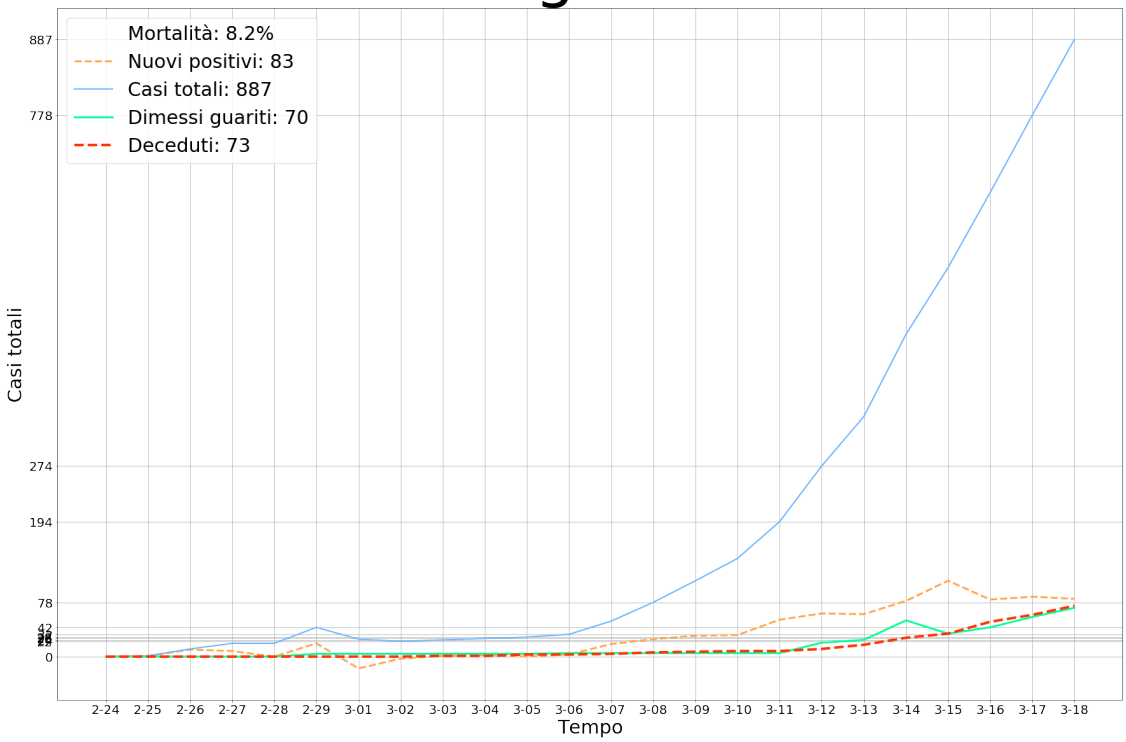
Friuli Venezia Giulia



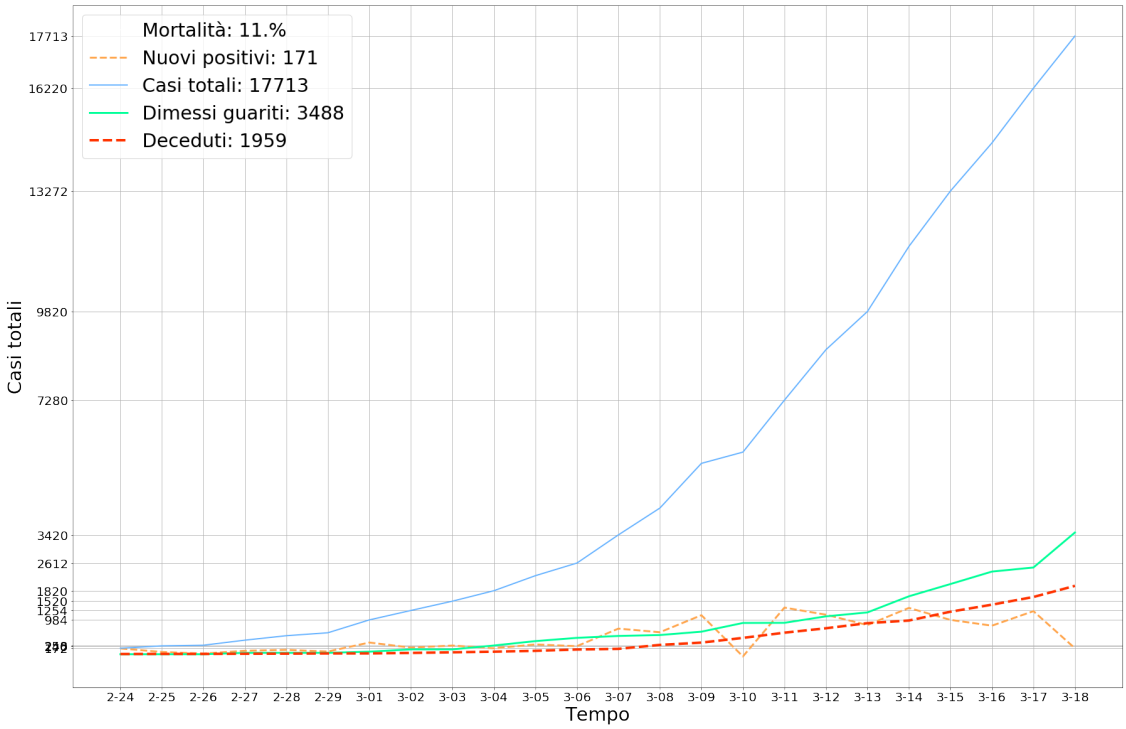
Lazio



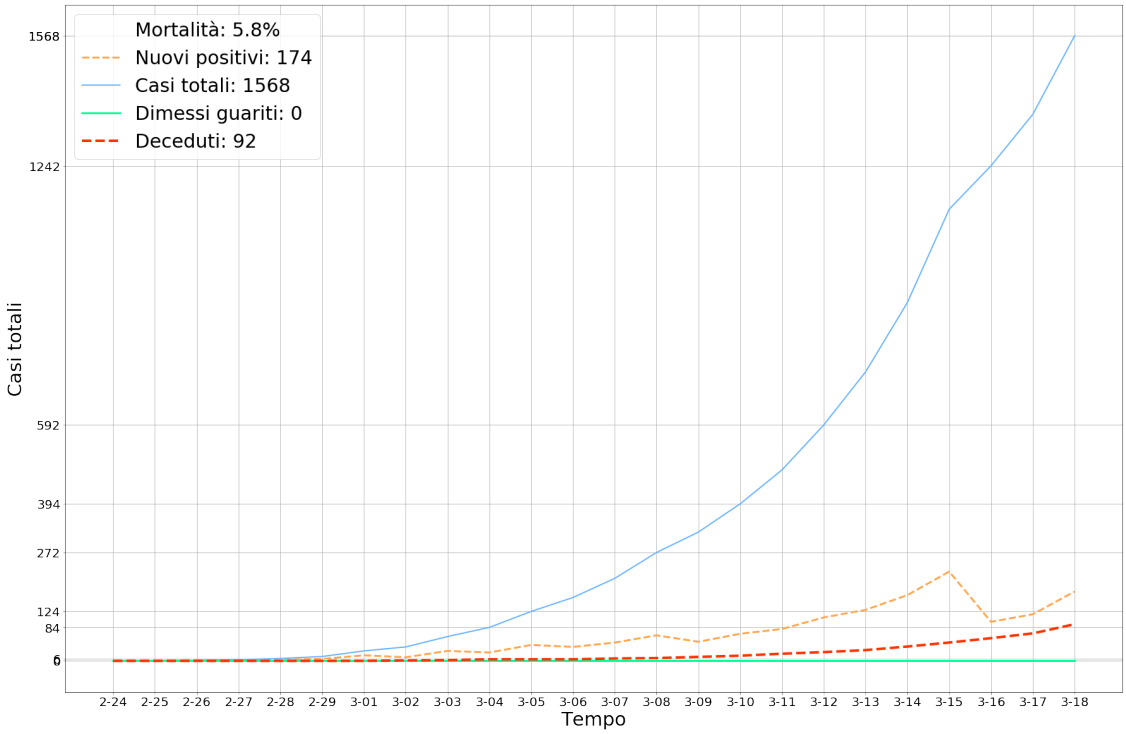
Liguria


Mortalità: 8.2%
Nuovi positivi: 83
Casi totali: 887
Dimessi guariti: 70
Deceduti: 73
887
778
274
194
78
42
0
Casi totali
2-24
2-25
2-26
2-27
2-28
2-29
3-01
3-02
3-03
3-04
3-05
3-06
3-07
3-08
3-09
3-10
3-11
3-12
3-13
3-14
3-15
3-16
3-17
3-18
Tempo

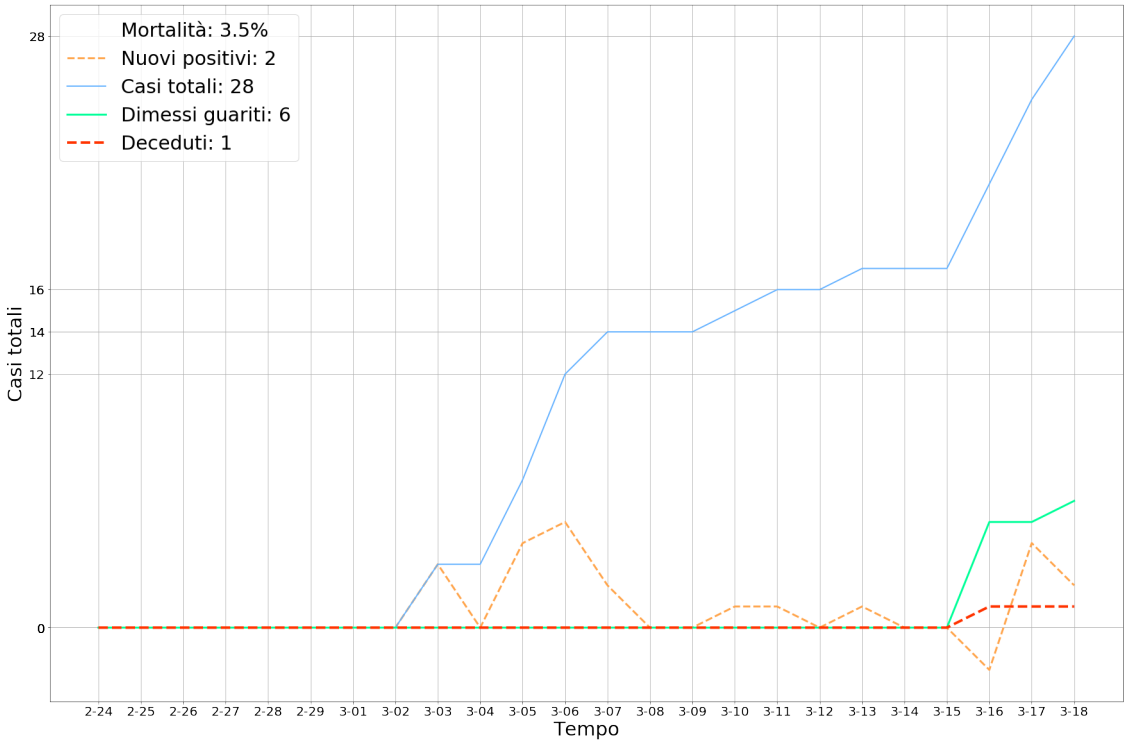
Lombardia



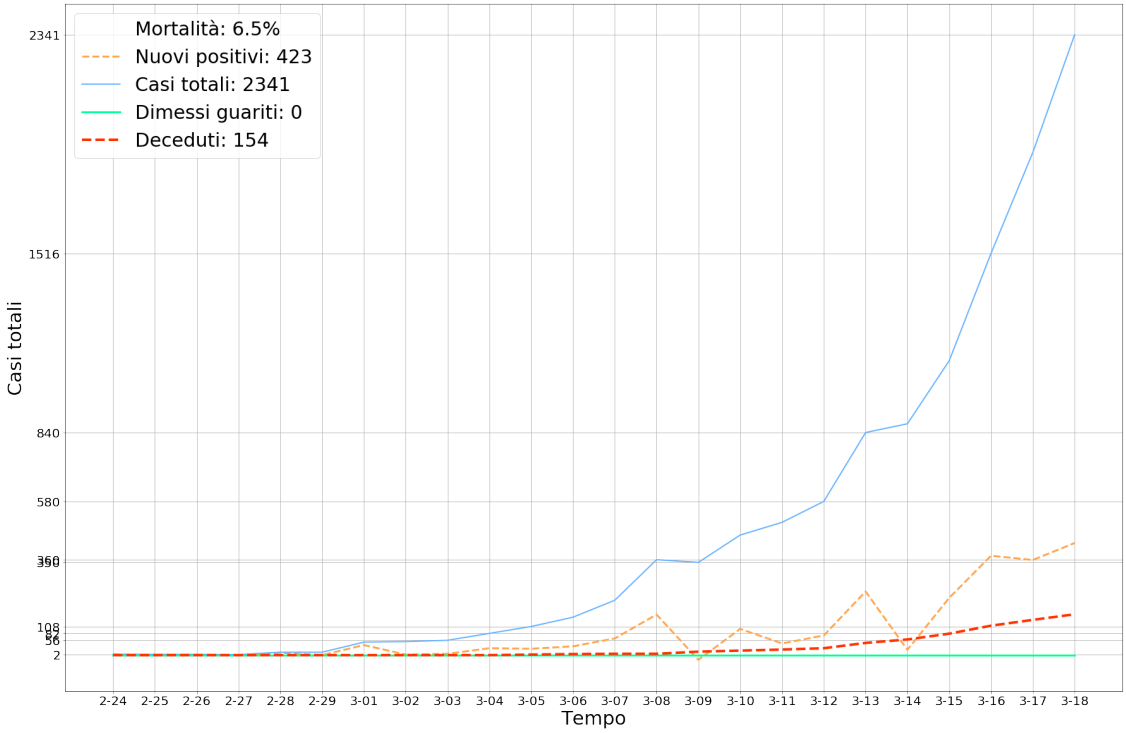
Marche



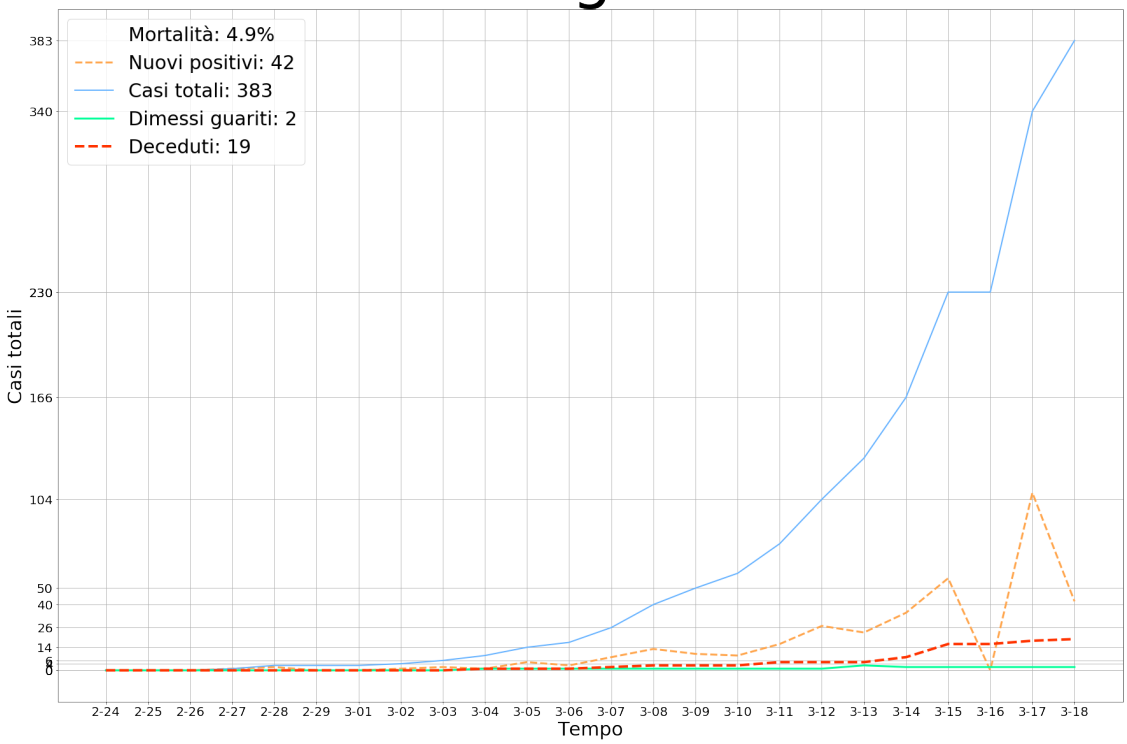
Molise


Mortalità: 3.5%
Nuovi positivi: 2
Casi totali: 28
Dimessi guariti: 6
Deceduti: 1
Casi totali
28
16
14
12
0
2-24
2-25
2-26
2-27
2-28
2-29
3-01
3-02
3-03
3-04
3-05
3-06
3-07
3-08
3-09
3-10
3-11
3-12
3-13
3-14
3-15
3-16
3-17
3-18
Tempo

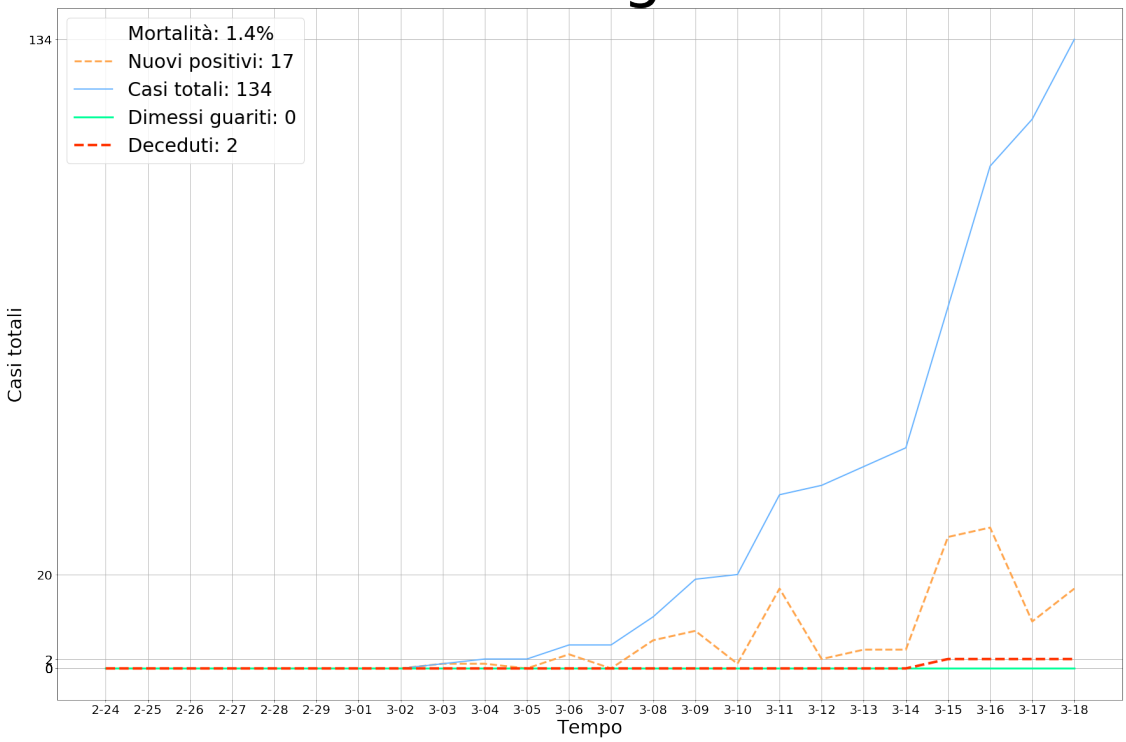
Piemonte



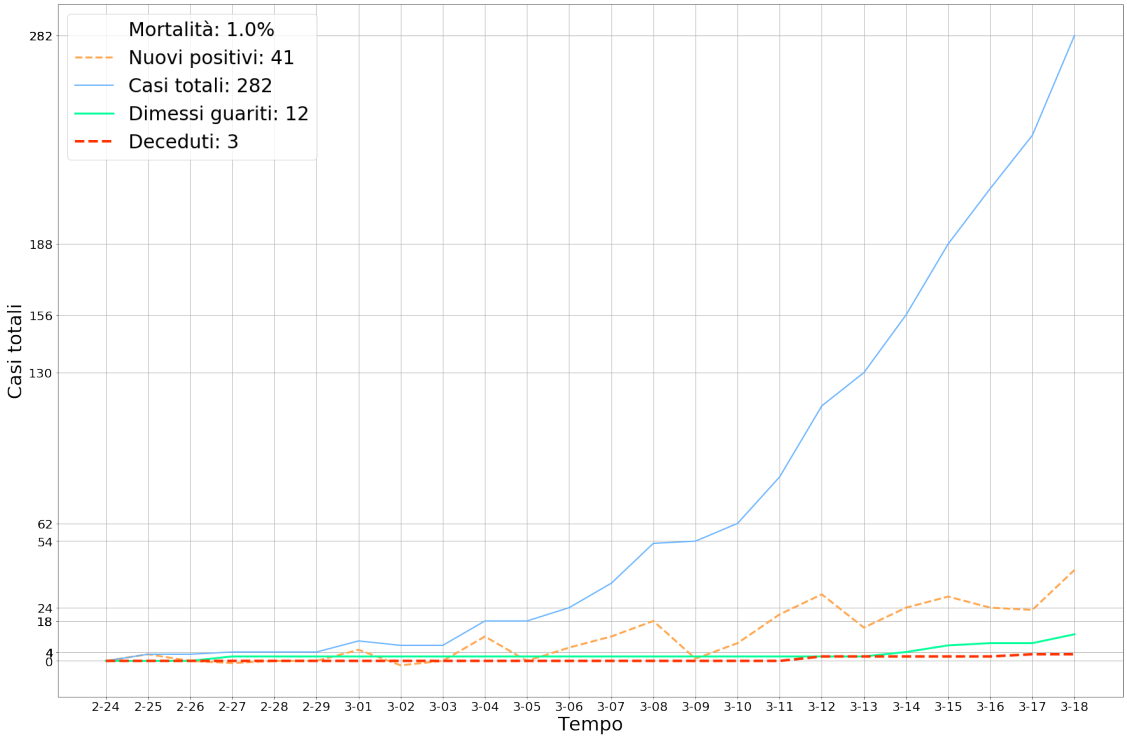
Puglia



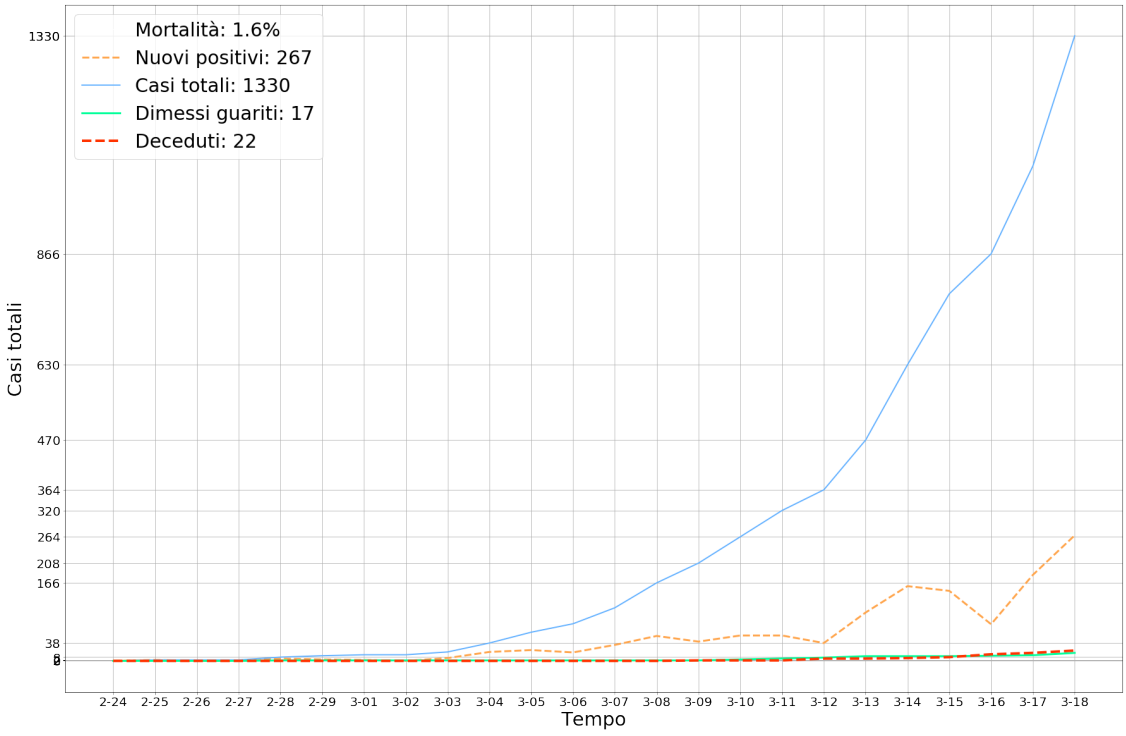
Sardegna


Mortalità: 1.4%
Nuovi positivi: 17
Casi totali: 134
Dimessi guariti: 0
Deceduti: 2
Casi totali
134
20
2
0
2-24
2-25
2-26
2-27
2-28
2-29
3-01
3-02
3-03
3-04
3-05
3-06
3-07
3-08
3-09
3-10
3-11
3-12
3-13
3-14
3-15
3-16
3-17
3-18
Tempo

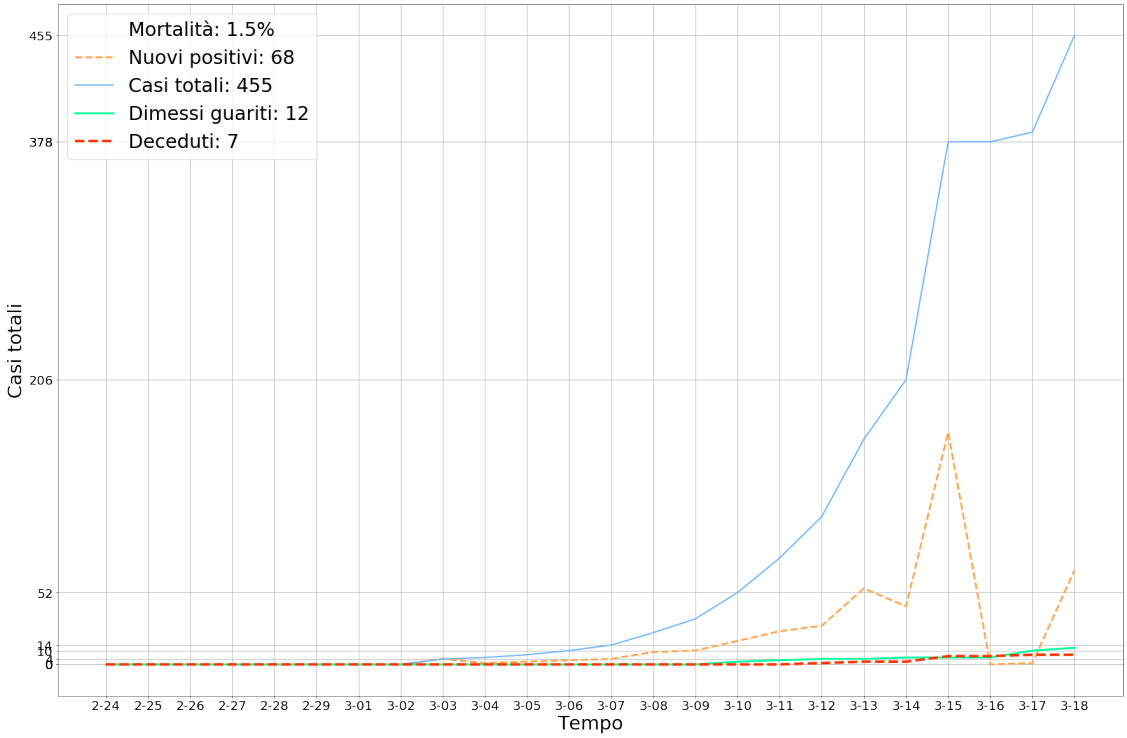
Sicilia



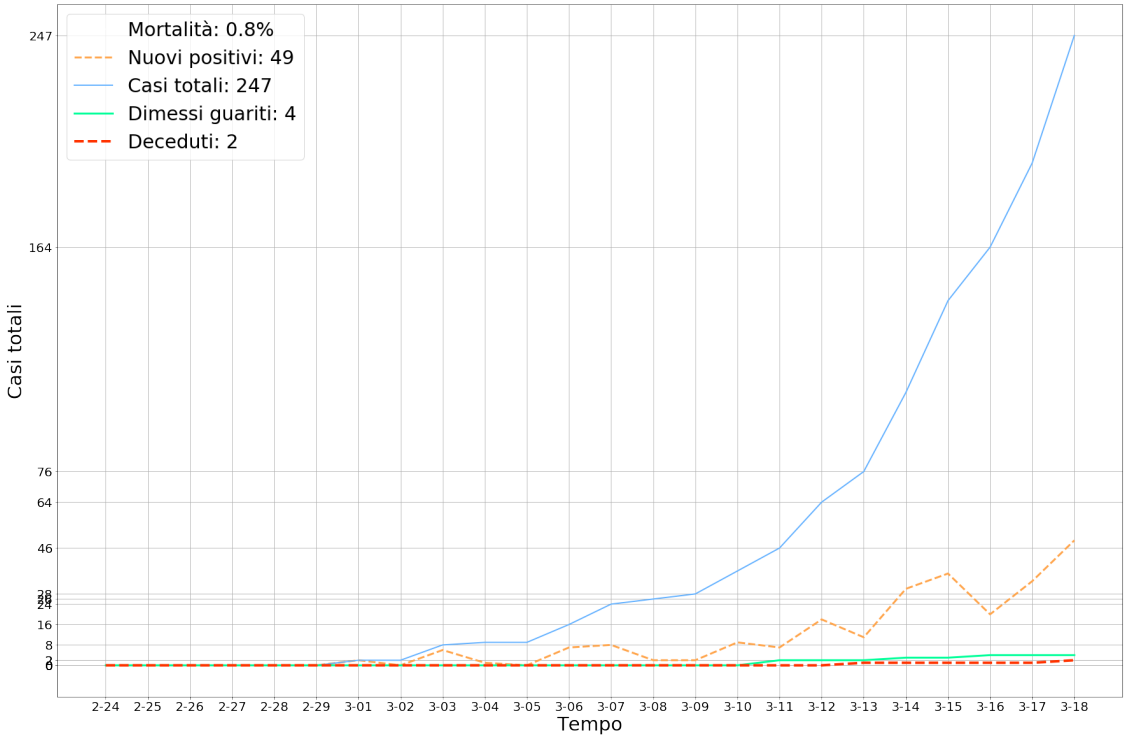
Toscana



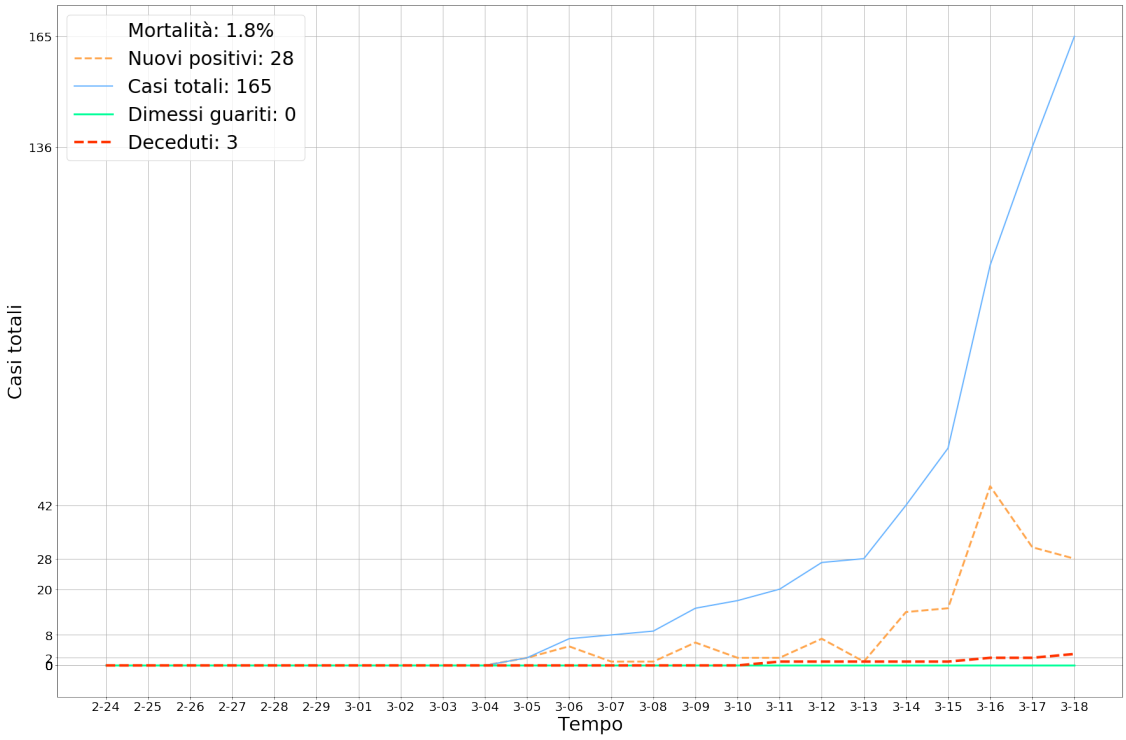
P.A. Trento

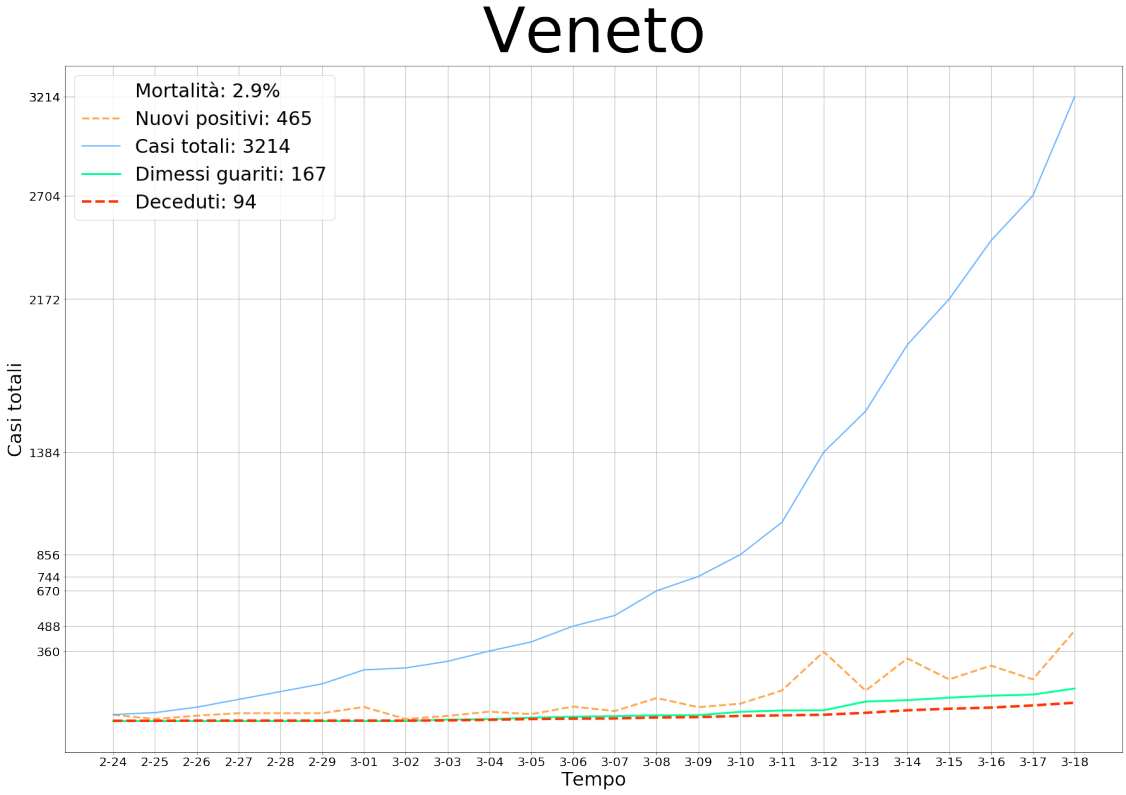

Mortalità: 1.5%
Nuovi positivi: 68
Casi totali: 455
Dimessi guariti: 12
Deceduti: 7
Casi totali
455
378
206
52
14
10
4
0
2-24
2-25
2-26
2-27
2-28
2-29
3-01
3-02
3-03
3-04
3-05
3-06
3-07
3-08
3-09
3-10
3-11
3-12
3-13
3-14
3-15
3-16
3-17
3-18
Tempo

Umbria



Valle d'Aosta






```
<Figure size 2160x1440 with 0 Axes>
```

[4]:
```
# dati-andamento-nazionale
df_nazionale = pd.read_csv('COVID-19/dati-andamento-nazionale/
 ↪dpc-covid19-ita-andamento-nazionale.csv')
```

[5]:
```
df_nazionale.columns
```

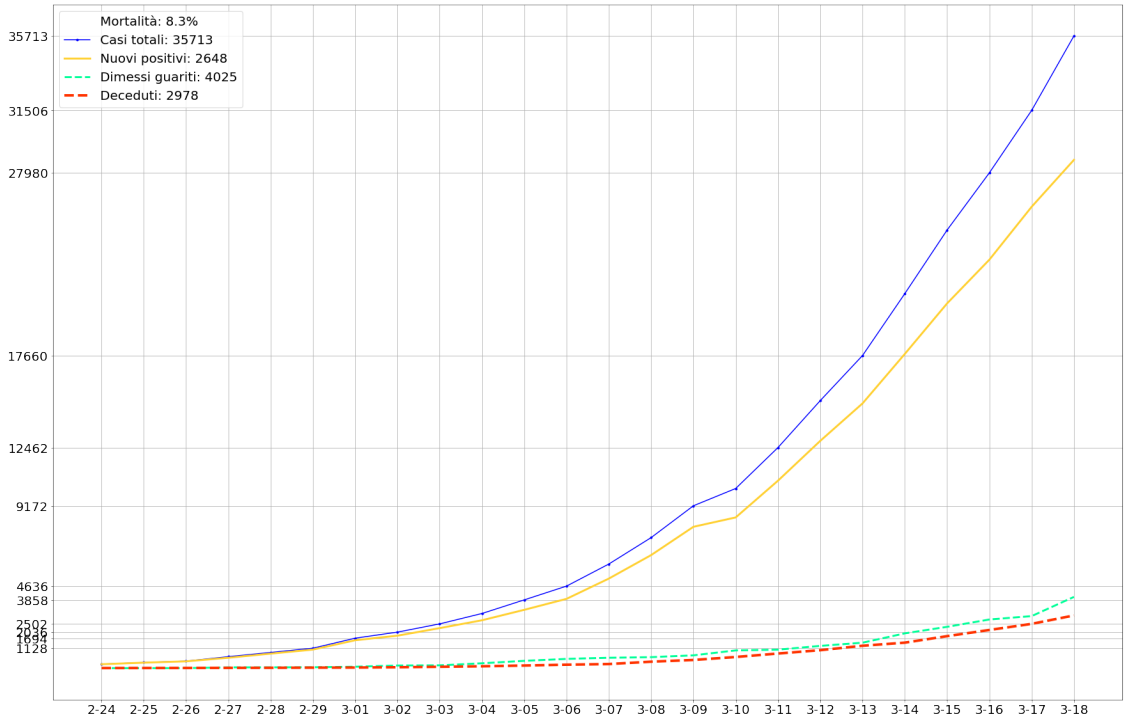
[5]:
```
Index(['data', 'stato', 'ricoverati_con_sintomi', 'terapia_intensiva',
       'totale_ospedalizzati', 'isolamento_domiciliare',
       'totale_attualmente_positivi', 'nuovi_attualmente_positivi',
       'dimessi_guariti', 'deceduti', 'totale_casi', 'tamponi'],
      dtype='object')
```

[6]:
```
x1 = df_nazionale.data
x = []
for f in x1:
    x.append(f[6:10])
```

```
totale_casi = df_nazionale.totale_casi

tot_nuovi_postivi = []
tot_deceduti = []
tot_guariti = []
casi_totali = []
ticks = []
ticks_1 = []

for p in totale_casi:
    casi_totali.append(p)

for w in totale_casi:
    if w % 2 == 0 and w > 1000:
        ticks.append(w)
    else:
        pass

for w1 in df_nazionale.nuovi_attualmente_positivi:
    tot_nuovi_postivi.append(w1)

for w2 in df_nazionale.dimessi_guariti:
    tot_guariti.append(w2)

for w3 in df_nazionale.deceduti:
    tot_deceduti.append(w3)



ticks_1.append(casi_totali[-1])
ticks.extend(ticks_1)



nuovi_positivi = df_nazionale.totale_attualmente_positivi
totale_deceduti = df_nazionale.deceduti
totale_guariti = df_nazionale.dimessi_guariti

plt.yticks(ticks)


plt.rc('ytick', labelsize=12)
plt.rc('xtick', labelsize=10)

plt.rcParams["figure.figsize"]=20,20

toll_1_tot = int(casi_totali[-1])
```

```
toll_2_tot = int(tot_deceduti[-1])

death_toll = (toll_2_tot/toll_1_tot)*100
conv_deth_toll = str(death_toll)

plt.plot(death_toll, color='#FFFFFF', label="Mortalità: {}%".
 ↪format(conv_deth_toll[:3]))
plt.plot(x, totale_casi, 'b.-',label='Casi totali: {}'.format(casi_totali[-1]))
plt.plot(x, nuovi_positivi, color='#FFD133', linewidth=3, label="Nuovi positivi:
 ↪ {}".format(tot_nuovi_postivi[-1]))
plt.plot(x, totale_guariti, color='#00ff99', linestyle="--", linewidth=3,␣
 ↪label='Dimessi guariti: {}'.format(tot_guariti[-1]))
plt.plot(x, totale_deceduti, color='#ff3300', linestyle="--", linewidth=4,␣
 ↪label='Deceduti: {}'.format(tot_deceduti[-1]))

plt.title("Andamento nazionale", fontsize=100)
plt.legend(prop={'size': 20})
plt.grid()



print(death_toll)
print(ticks)
```

```
8.338700193206956
[1128, 1694, 2036, 2502, 3858, 4636, 9172, 12462, 17660, 27980, 31506, 35713]
```



1.1 Inizo modello

[7]:
```
fcst = []
fcst_inv = []
fcst_inv_disp = []
fcst_inv_pari = []

x1 = df_nazionale.data
x2 = []
for f in x1:
    x2.append(f[6:10])

for w in range(len(tot_nuovi_postivi)):
    fcst.append(w)

for m in fcst:
    if m != 1 and m != 0:
        fcst_inv_pari.append(m * (-1))
    else:
        pass

for t in fcst:
```

```
    if t != 0:
        fcst_inv_disp.append(t * (-1))


res = []
res3 = []
for q,l in zip(fcst_inv_disp, fcst_inv_pari):

    f = int(tot_nuovi_postivi[q])
    z = int(tot_nuovi_postivi[l])
    res1 = f - z
    res2 = (f/z)*100
    res.append(res1)
    res3.append(res2)

model = []
last = []

for ext in tot_nuovi_postivi:
    if ext % 2 == 0:
        model.append(ext)
    else:
        pass

last.append(tot_nuovi_postivi[-1])
model.extend(last)

print(model)

adapt = x[2:]
print(res3)


plt.yticks(model)



plt.plot(adapt, res, label="Nuovi infetti")
plt.plot(adapt, res3, label= "% Variazione")
plt.legend()
plt.grid()
plt.show()
```

```
[90, 74, 228, 528, 258, 428, 590, 620, 1326, 1598, 2076, 2116, 2470, 2648, 2648]
[88.59150217464034, 121.01214574898786, 86.57553452506134, 102.07513416815743,
132.08884688090737, 94.086260560249, 108.33333333333333, 392.43856332703217,
33.103879849812266, 120.51282051282051, 115.80786026200873, 184.6774193548387,
```

```
105.08474576271188, 133.18284424379232, 103.50467289719627, 165.89147286821705,
48.86363636363637, 231.57894736842107, 97.85407725321889, 114.77832512315271,
274.3243243243243, 82.22222222222221]
```

[ ]: