

```
In [24]: # Dopo aver svolto le operazioni di pulizia dei dataset (una tantum), leg

import numpy as np
import pandas as pd

comuni_clean = pd.read_csv("/Users/michelavenier/Desktop/EPICODE/BUILD_W
comuni_clean=comuni_clean.drop('Unnamed: 0', axis=1)

province_clean = pd.read_csv("/Users/michelavenier/Desktop/EPICODE/BUILD_
province_clean=province_clean.drop('Unnamed: 0', axis=1)

regioni_clean = pd.read_csv("/Users/michelavenier/Desktop/EPICODE/BUILD_W
regioni_clean=regioni_clean.drop('Unnamed: 0', axis=1)

ripartizioni_clean = pd.read_csv("/Users/michelavenier/Desktop/EPICODE/BU
ripartizioni_clean=ripartizioni_clean.drop('Unnamed: 0', axis=1)

#stampo i dataset importati per verificare che siano formattati correttam
print("comuni",comuni_clean['Regione'].unique(), comuni_clean.iloc[161:17

comuni ['Piemonte' ' ' "Valle d'Aosta" 'Lombardia' 'Trentino-Alto Adige/Su
dtirol'
'Veneto' 'Friuli-Venezia Giulia' 'Liguria' 'Emilia-Romagna' 'Toscana'
'Umbria' 'Marche' 'Lazio' 'Abruzzo' 'Molise' 'Campania' 'Puglia'
'Basilicata' 'Calabria' 'Sicilia' 'Sardegna']      Denominazione Unnamed
: 1  Regione Sigla automobilistica Popolazione2011
161      Nichelino      Piemonte      TO      4
7851
162      Noasca      Piemonte      TO
169
163      Nole      Piemonte      TO
6910
164      Nomaglio      Piemonte      TO
312
165      None      Piemonte      TO
7998
166      Novalesa      Piemonte      TO
560
167      Oglianico      Piemonte      TO
1426
168      Orbassano      Piemonte      TO      2
2537
169  Orio Canavese      Piemonte      TO
829

province      SNo      Date Country      RegionCode RegionName      Pro
vinceCode \
0      0      2020-02-24 18:00:00      ITA      13      Abruzzo      66
1      1      2020-02-24 18:00:00      ITA      13      Abruzzo      67

ProvinceName ProvinceAbbreviation      Latitude      Longitude      TotalPositiveC
ases
0      L'Aquila      AQ      42.351222      13.398438
```

```

0
1      Teramo      TE  42.658918  13.704400
0

regioni      SNo      Date Country  RegionCode  RegionName  La
titude  \
0      0  2020-02-24 18:00:00  ITA      13      Abruzzo  42.351222
1      1  2020-02-24 18:00:00  ITA      17  Basilicata  40.639471

      Longitude  HospitalizedPatients  IntensiveCarePatients  \
0  13.398438      0      0
1  15.805148      0      0

      TotalHospitalizedPatients  HomeConfinement  CurrentPositiveCases  \
0      0      0      0
1      0      0      0

      NewPositiveCases  Recovered  Deaths  TotalPositiveCases  TestsPerforme
d
0      0      0      0      0      Na
N
1      0      0      0      0      Na
N

ripartizioni      Codice Regione Ripartizione geografica      Regione
0      1      Nord-ovest      Piemonte
1      2      Nord-ovest  Valle d'Aosta

```

```

In [25]: # Visualizzazione dei Contagi massimi per regione sulla mappa d'Italia

import matplotlib.pyplot as plt
import geopandas as gpd
italy = gpd.read_file("/Users/michelavenier/Desktop/EPICODE/BUILD_WEEK_2/

regioni_somma_contagi = regioni_clean.groupby('RegionName')['TotalPositiv
print("regioni_somma_contagi", regioni_somma_contagi)

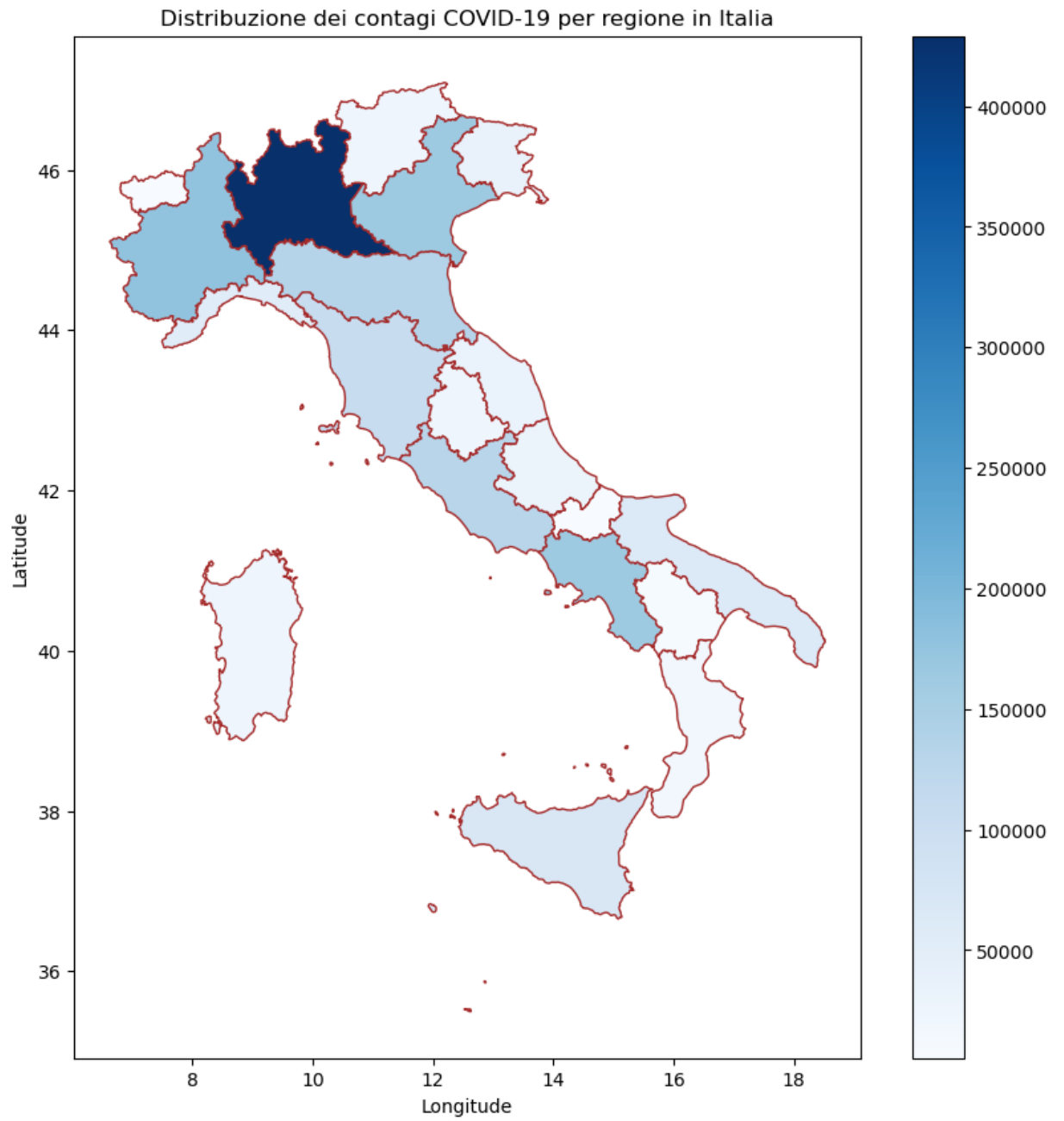
italy_with_cases = italy.merge(regioni_somma_contagi, left_on='name', rig

italy_with_cases.plot(column='TotalPositiveCases', cmap='Blues', edgecolo

plt.title('Distribuzione dei contagi COVID-19 per regione in Italia')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()

```

regioni_somma_contagi		RegionName	TotalPositiveCase
s			
0	Abruzzo	30274	
1	Basilicata	8940	
2	Calabria	18537	
3	Campania	165293	
4	Emilia-Romagna	133761	
5	Friuli-Venezia Giulia	35467	
6	Lazio	130255	
7	Liguria	53779	
8	Lombardia	429109	
9	Marche	32393	
10	Molise	5286	
11	Piemonte	177788	
12	Puglia	64341	
13	Sardegna	24186	
14	Sicilia	71489	
15	Toscana	108397	
16	Trentino-Alto Adige/Sudtirolo	25361	
17	Umbria	25144	
18	Valle d'Aosta	6726	
19	Veneto	165249	



```
In [26]: # Visualizzazione dei Decessi massimi per regione sulla mappa d'Italia

import matplotlib.pyplot as plt
import geopandas as gpd

italy = gpd.read_file("/Users/michelavenier/Desktop/EPICODE/BUILD_WEEK_2/
#da sito web https://cartographyvectors.com/map/357-italy-with-regions

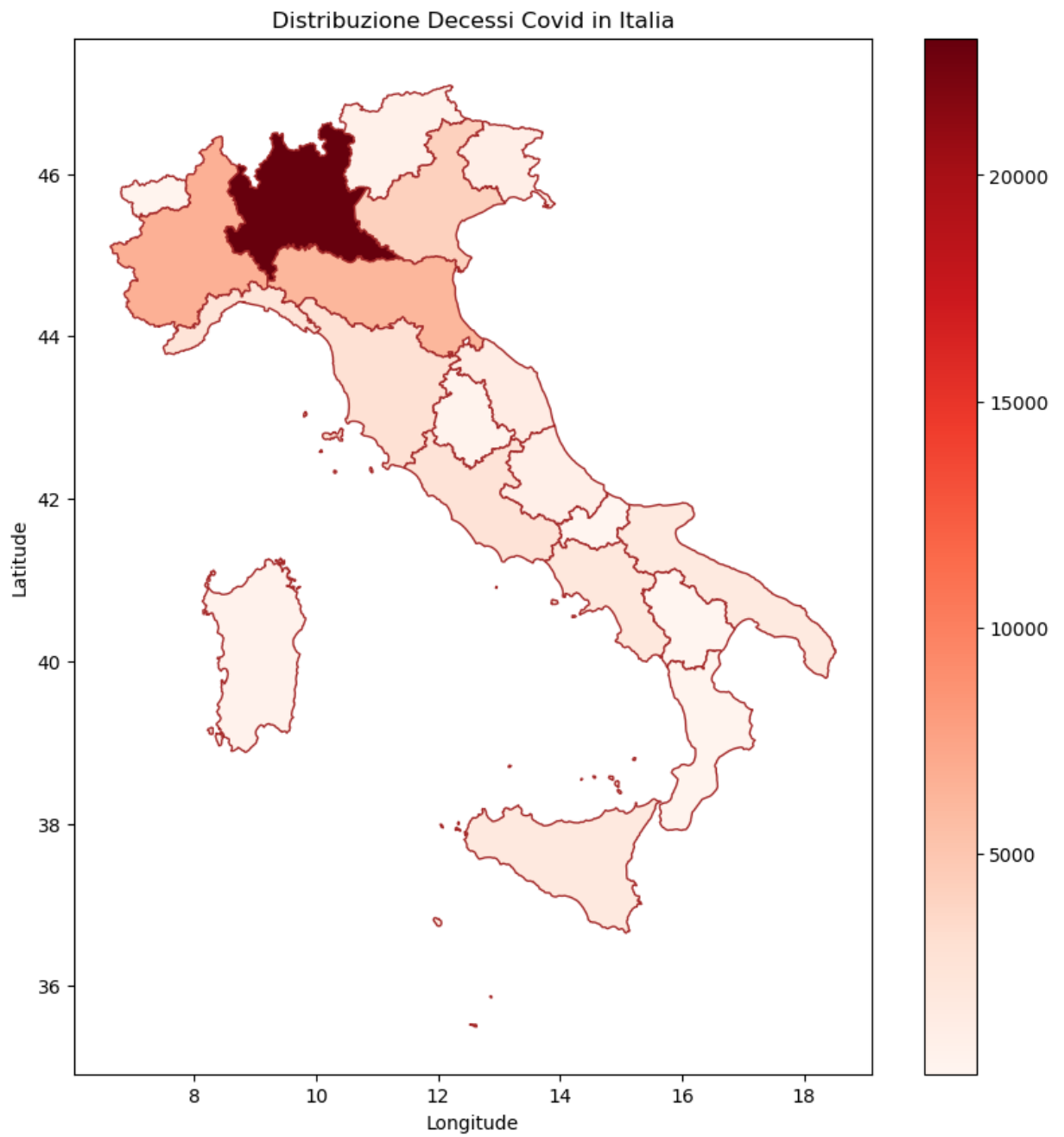
regioni_decessi = regioni_clean.groupby('RegionName')['Deaths'].max().res
print("regioni_decessi", regioni_decessi)

italy_with_cases = italy.merge(regioni_decessi, left_on='name', right_on=

italy_with_cases.plot(column='Deaths', cmap='Reds', edgecolor='brown', le

plt.title('Distribuzione Decessi Covid in Italia')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

regioni_decessi		RegionName	Deaths
0	Abruzzo	980	
1	Basilicata	175	
2	Calabria	343	
3	Campania	1990	
4	Emilia-Romagna	6162	
5	Friuli-Venezia Giulia	1035	
6	Lazio	2622	
7	Liguria	2521	
8	Lombardia	23024	
9	Marche	1327	
10	Molise	135	
11	Piemonte	6623	
12	Puglia	1712	
13	Sardegna	531	
14	Sicilia	1759	
15	Toscana	2867	
16	Trentino-Alto Adige/Sudtirolo	691	
17	Umbria	460	
18	Valle d'Aosta	333	
19	Veneto	4210	



```
In [27]: # Visualizzazione dei Pazienti Guariti massimi per regione sulla mappa d'

import matplotlib.pyplot as plt
import geopandas as gpd

italy = gpd.read_file("/Users/michelavenier/Desktop/EPICODE/BUILD_WEEK_2/
#da sito web https://cartographyvectors.com/map/357-italy-with-regions

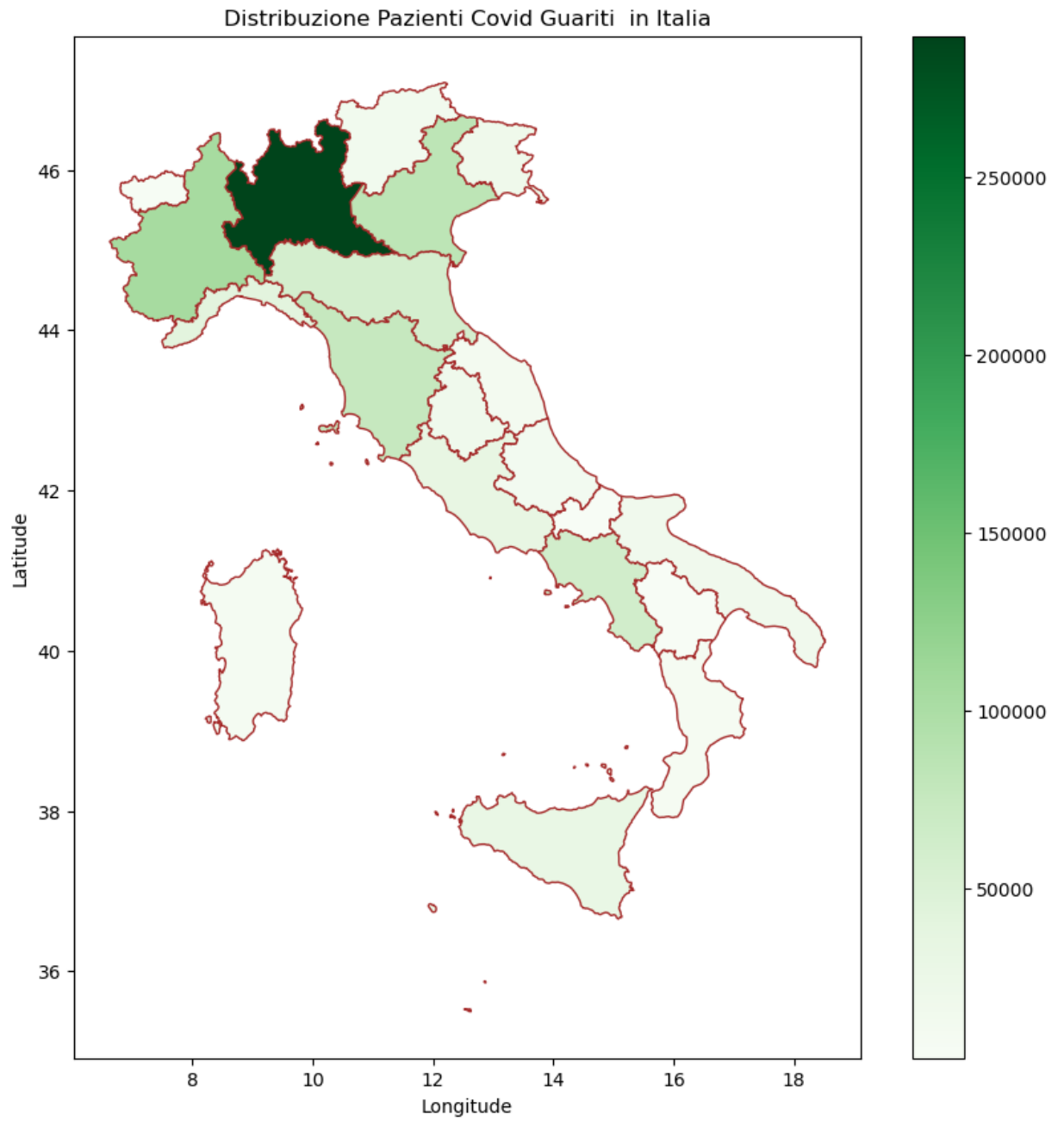
regioni_guariti = regioni_clean.groupby('RegionName')['Recovered'].max().
print("regioni_guariti", regioni_guariti)

italy_with_cases = italy.merge(regioni_guariti, left_on='name', right_on=

italy_with_cases.plot(column='Recovered', cmap='Greens', edgecolor='brown

plt.title('Distribuzione Pazienti Covid Guariti in Italia')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

regioni_guariti		RegionName	Recovered
0	Abruzzo	12206	
1	Basilicata	2397	
2	Calabria	7533	
3	Campania	62060	
4	Emilia-Romagna	59432	
5	Friuli-Venezia Giulia	19401	
6	Lazio	33422	
7	Liguria	40842	
8	Lombardia	289706	
9	Marche	11008	
10	Molise	2438	
11	Piemonte	105127	
12	Puglia	16795	
13	Sardegna	8695	
14	Sicilia	29984	
15	Toscana	76331	
16	Trentino-Alto Adige/Sudtirolo	13995	
17	Umbria	18619	
18	Valle d'Aosta	5406	
19	Veneto	84235	





```
In [28]: # prime 3 regioni per pazienti ospedalizzati

HospitalizedPatients_by_region = regioni_clean.groupby('RegionName')['Tot
sorted_regions_HospitalizedPatients = HospitalizedPatients_by_region.sort
top_3_regions_HospitalizedPatients = sorted_regions_HospitalizedPatients.
print("Le tre regioni con il maggior tasso di pazienti guariti sono:")
print(top_3_regions_HospitalizedPatients)
region_names = top_3_regions_HospitalizedPatients.index
total_hospitalized_counts = top_3_regions_HospitalizedPatients.values

plt.figure(figsize=(8, 5))
plt.gca().set_facecolor('#D8D6CD')
plt.bar(region_names, total_hospitalized_counts, color='lightblue', edgec
plt.xlabel('Regioni')
plt.ylabel('Totale Pazienti Ospedalizzati')
plt.title('Top 3 Regioni per Pazienti Ospedalizzati in Italia')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

Le tre regioni con il maggior tasso di pazienti guariti sono:

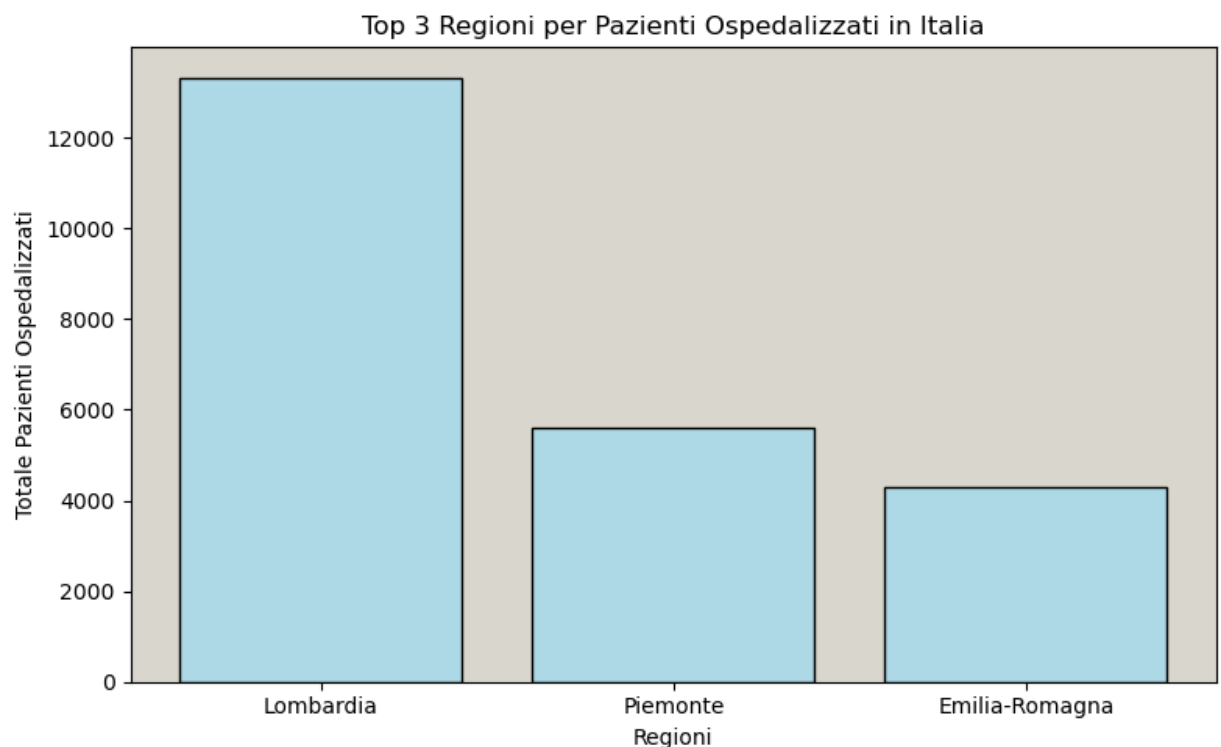
RegionName

Lombardia 13328

Piemonte 5618

Emilia-Romagna 4310

Name: TotalHospitalizedPatients, dtype: int64



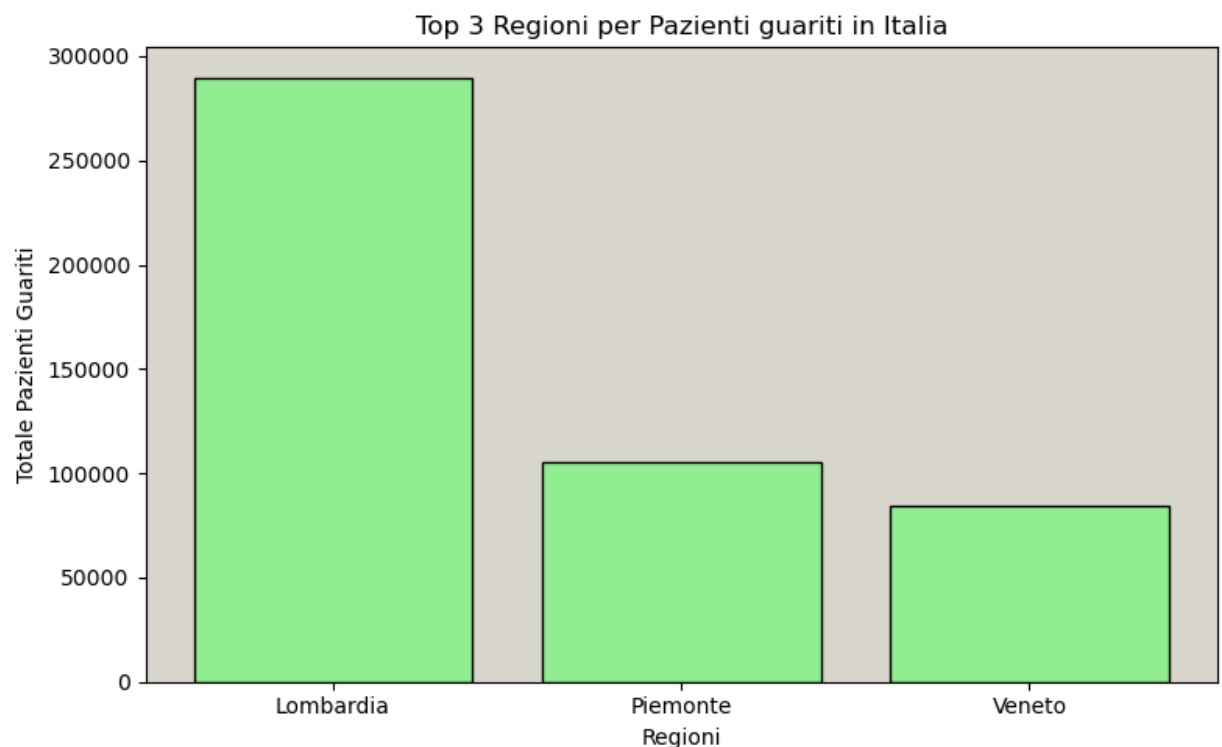
```
In [29]: # prime 3 regioni per numero di pazienti guariti

total_Recovered_by_region = regioni_clean.groupby('RegionName')['Recovere
sorted_regions = total_Recovered_by_region.sort_values(ascending=False)
top_3_regions = sorted_regions.head(3)
print("Le tre regioni con il maggior tasso di pazienti guariti sono:")
print(top_3_regions)
region_names = top_3_regions.index
total_hospitalized_counts = top_3_regions.values

plt.figure(figsize=(8, 5))
plt.gca().set_facecolor('#D8D6CD')
plt.bar(region_names, total_hospitalized_counts, color='lightgreen', edgec
plt.xlabel('Regioni')
plt.ylabel('Totale Pazienti Guariti')
plt.title('Top 3 Regioni per Pazienti guariti in Italia')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

Le tre regioni con il maggior tasso di pazienti guariti sono:

```
RegionName
Lombardia      289706
Piemonte       105127
Veneto         84235
Name: Recovered, dtype: int64
```

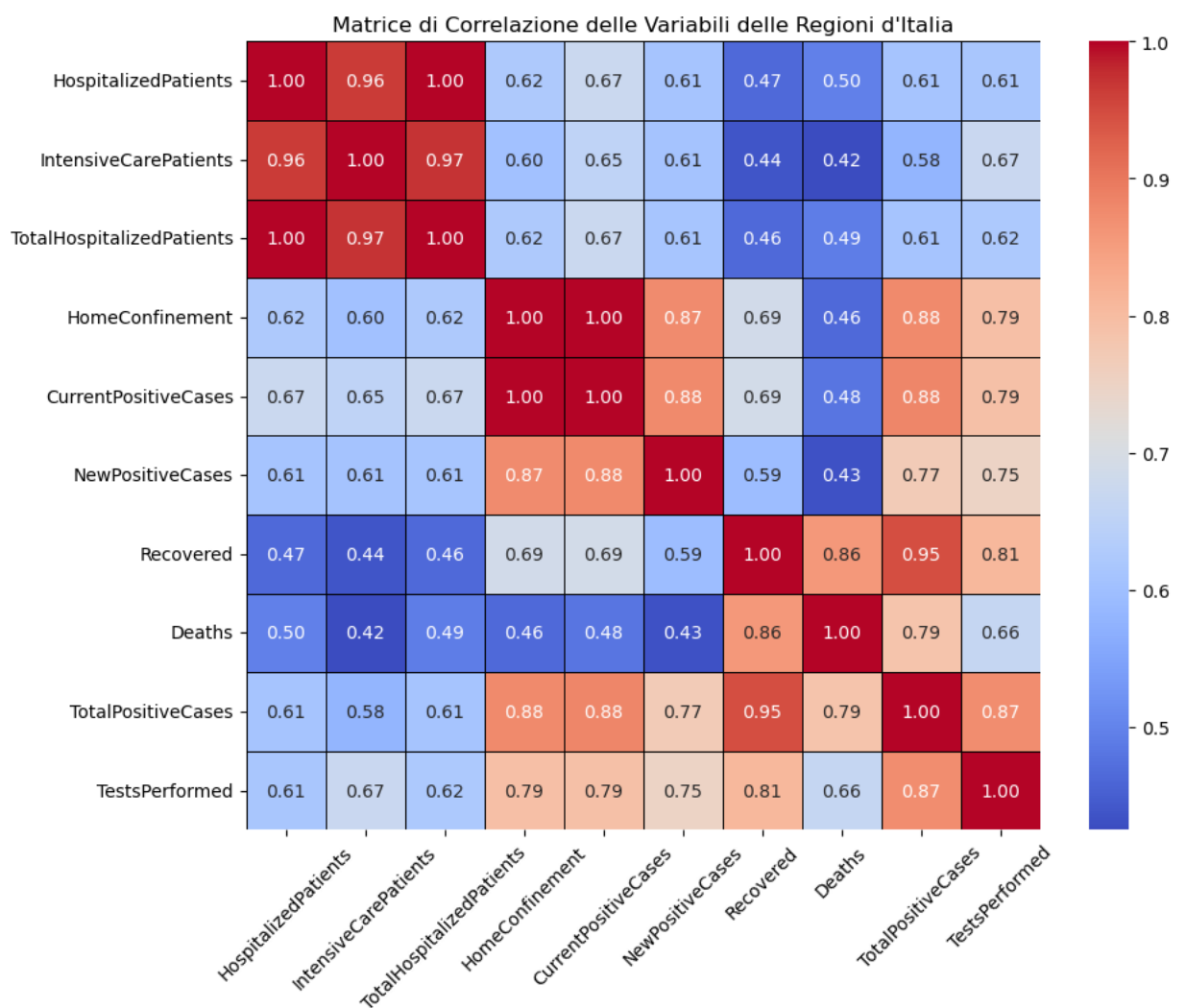


```
In [30]: # matrice di correlazione delle varie colonne del dataset regioni_clean
import seaborn as sns

# Selezione solo le colonne numeriche
colonne_numeriche = regioni_clean.select_dtypes(include='number')

# Calcolo la matrice di correlazione
corr_matrix = colonne_numeriche.corr()
corr_matrix = corr_matrix.iloc[4:, 4:]

#raffiguro il grafico heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidth=1)
plt.title('Matrice di Correlazione delle Variabili delle Regioni d'Italia')
plt.xticks(rotation=45)
plt.show()
```



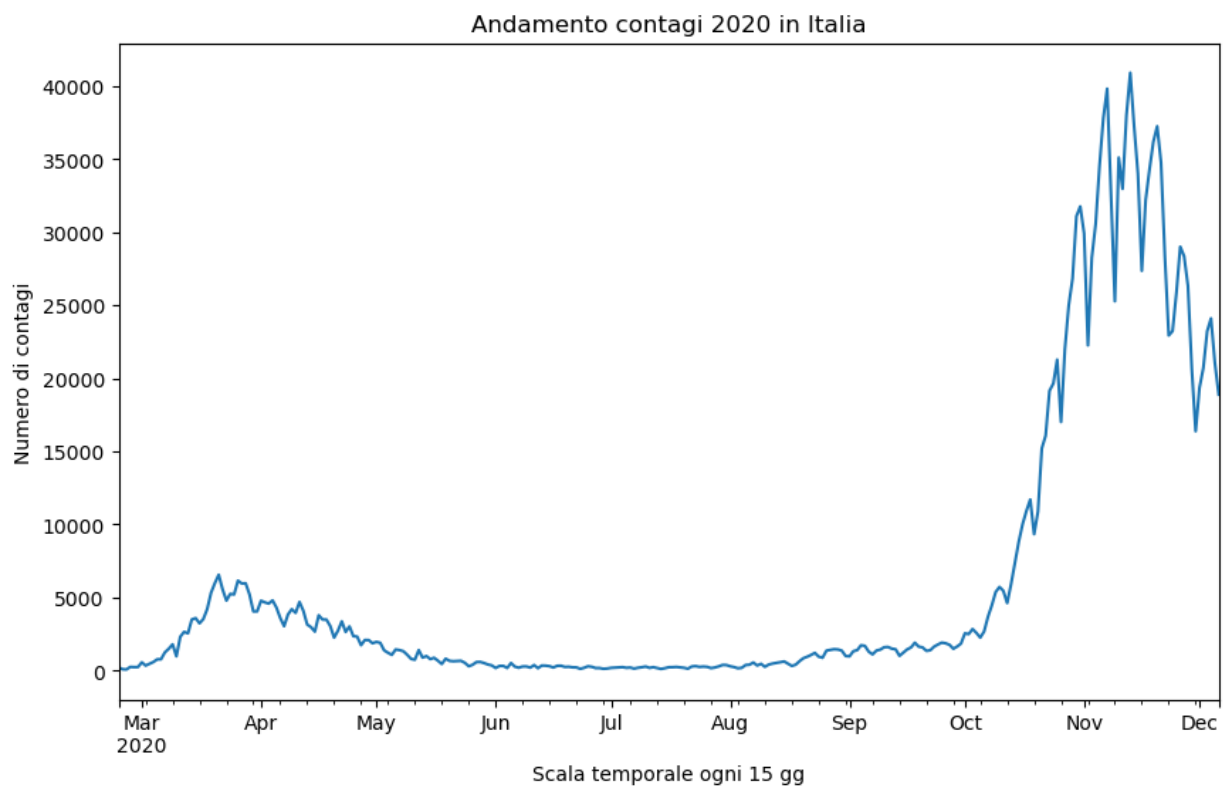
```
In [31]: # evoluzione del contagio nell'arco del 2020, tra febbraio e dicembre

# mi assicuro che la colonna "Date" sia visualizzata come una data e la i
regioni_clean = pd.read_csv("/Users/michelavenier/Desktop/EPICODE/BUILD_W
regioni_clean=regioni_clean.drop('Unnamed: 0', axis=1)
regioni_clean['Date'] = pd.to_datetime(regioni_clean['Date'])
regioni_clean.set_index('Date', inplace=True)

# visualizzo il grafico a linea
c = regioni_clean.resample('D')['NewPositiveCases'].sum().plot(kind="line
print("Totale_contagiati_2020",regioni_clean['NewPositiveCases'].sum())

plt.title('Andamento contagi 2020 in Italia')
plt.xlabel('Scala temporale ogni 15 gg')
plt.ylabel('Numero di contagi')
plt.show()
```

Totale\_contagiati\_2020 1729061



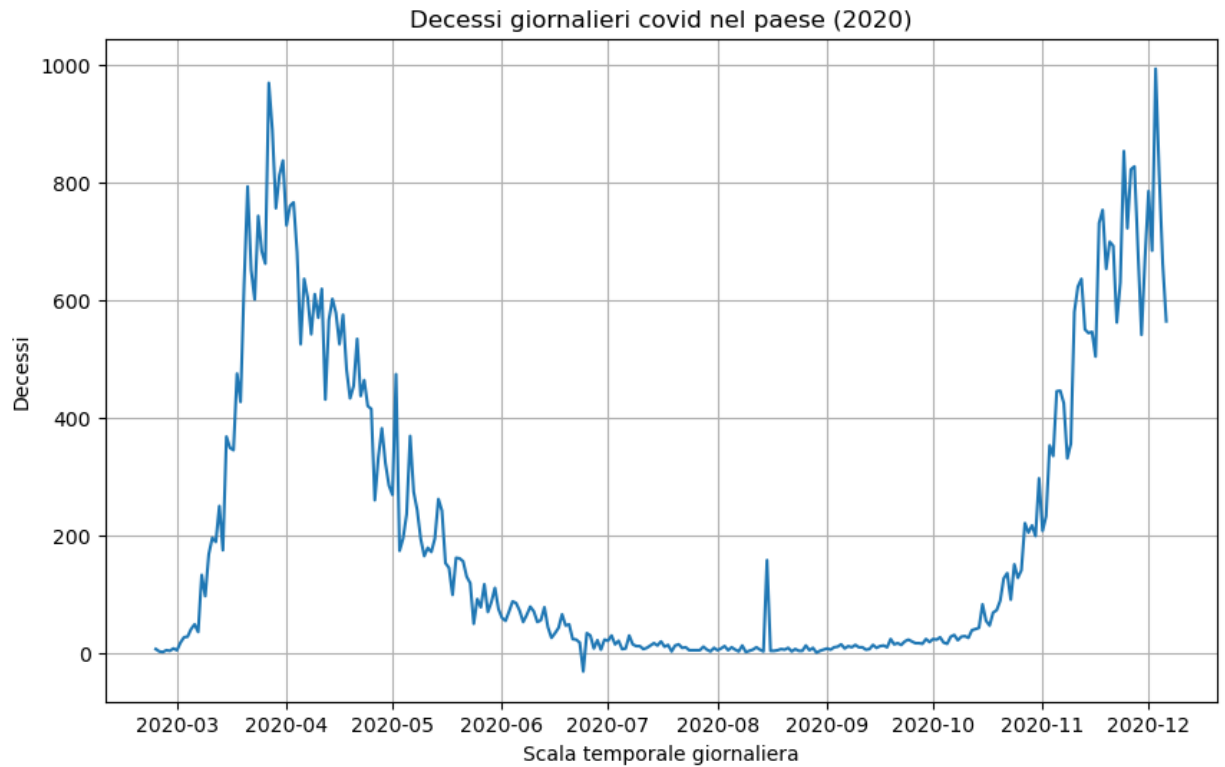
```
In [32]: # Calcolo e visualizzazione della Mortalità Giornaliera per Covid-19 tra

contagimorti_giornalieri=regioni_clean.resample('D')[['NewPositiveCases',
contagimorti_giornalieri['nuovi_morti']=0
for i in range(0, 287):
    if i==0:
        contagimorti_giornalieri.iloc[0,3]=contagimorti_giornalie
    else:
        contagimorti_giornalieri.iloc[i,3]=contagimorti_giornalie

print(contagimorti_giornalieri)
#print(contagimorti_giornalieri[['Date', 'TotalPositiveCases', 'Deaths']])
#contagimorti_giornalieri['letalita']=(contagimorti_giornalieri['Deaths']
#letalita=contagimorti_giornalieri['letalita'].round(decimals=3)
plt.figure(figsize=(10, 6))
plt.plot(contagimorti_giornalieri.iloc[:,0],contagimorti_giornalieri.iloc
plt.grid(True)
plt.title('Decessi giornalieri covid nel paese (2020)')
plt.xlabel('Scala temporale giornaliera')
plt.ylabel('Decessi')
plt.show()
```

	Date	NewPositiveCases	Deaths	nuovi_morti
0	2020-02-24	221	7	7
1	2020-02-25	93	10	3
2	2020-02-26	78	12	2
3	2020-02-27	250	17	5
4	2020-02-28	238	21	4
..	...	...	...	...
282	2020-12-02	20709	57045	684
283	2020-12-03	23225	58038	993
284	2020-12-04	24099	58852	814
285	2020-12-05	21052	59514	662
286	2020-12-06	18887	60078	564

[287 rows x 4 columns]



```
In [33]: # Per esercizio personale, abbiamo scritto una funzione che prende in input
# questa funzione è stata sviluppata dal seguente codice:
# Regione = input("Inserisci una Regione: ")
# comuni_regione = comuni_clean[comuni_clean['Regione'].str.contains(Regione)]
# print(f"I comuni della regione '{Regione}' sono: {comuni_regione}")

def comuni_per_regione(regione):
    comuni_regione = comuni_clean[comuni_clean['Regione'].str.contains(regione)]
    return comuni_regione

comuni_di_interesse = comuni_per_regione("Molise")
print(comuni_di_interesse)
```

['Acquaviva Collecroce', 'Baranello', 'Bojano', 'Bonefro', 'Busso', 'Campobasso', 'Campochiaro', 'Campodipietra', 'Campolieto', 'Campomarino', 'Casacalenda', 'Casalciprano', 'Castelbottaccio', 'Castellino del Biferno', 'Castelmauro', 'Castropignano', 'Cercemaggiore', 'Cercepisciolina', 'Civitacampomaro', 'Colle d'Anchise', 'Colletorto', 'Duronio', 'Ferrazzano', 'Fossalto', 'Gambatesa', 'Gildone', 'Guardiafiera', 'Guardiaregia', 'Guglionesi', 'Jelsi', 'Larino', 'Limosano', 'Lucito', 'Lupara', 'Macchia Valfortore', 'Mafalda', 'Matrice', 'Mirabello Sannitico', 'Molise', 'Monacilioni', 'Montagano', 'Montecilfone', 'Montefalcone nel Sannio', 'Montelongo', 'Montemitro', 'Montenero di Bisaccia', 'Montorio nei Frentani', 'Morrone del Sannio', 'Oratino', 'Palata', 'Petacciato', 'Petrella Tifernina', 'Pietracatella', 'Pietracupa', 'Portocannone', 'Provvidenti', 'Riccia', 'Ripabottoni', 'Ripalimosani', 'Roccavivara', 'Rotello', 'Salcito', 'San Biase', 'San Felice del Molise', 'San Giacomo degli Schiavoni', 'San Giovanni in Galdo', 'San Giuliano del Sannio', 'San Giuliano di Puglia', 'San Martino in Pensilis', 'San Massimo', 'San Polo Matese', 'Santa Croce di Magliano', 'Sant'Angelo Limosano', 'Sant'Elia a Pianisi', 'Sepino', 'Spinete', 'Tavenna', 'Termoli', 'Torella del Sannio', 'Toro', 'Trivento', 'Tufara', 'Ururi', 'Vinchiaturo', 'Acquaviva d'Isernia', 'Agnone', 'Bagnoli del Trigno', 'Belmonte del Sannio', 'Cantalupo nel Sannio', 'Capracotta', 'Carovilli', 'Carpinone', 'Castel del Giudice', 'Castelpetroso', 'Castelpizzuto', 'Castel San Vincenzo', 'Castelverrino', 'Cerro al Volturno', 'Chiauci', 'Civitanova del Sannio', 'Colli a Volturno', 'Conca Casale', 'Filignano', 'Forlì del Sannio', 'Fornelli', 'Frosolone', 'Isernia', 'Longano', 'Macchia d'Isernia', 'Macchiagodena', 'Miranda', 'Montaquila', 'Montenero Val Cocchiara', 'Monteroduni', 'Pesche', 'Pescolanciano', 'Pescopennataro', 'Pettoranello del Molise', 'Pietrabbondante', 'Pizzone', 'Poggio Sannita', 'Pozzilli', 'Rionero Sannitico', 'Roccamandolfi', 'Roccasicura', 'Rocchetta a Volturno', 'San Pietro Avellana', 'Sant'Agapito', 'Santa Maria del Molise', 'Sant'Angelo del Pesco', 'Sant'Elena Sannita', 'Scapoli', 'Sessano del Molise', 'Sesto Campano', 'Vastogirardi', 'Venafro']

```
In [34]: # calcolo delle 10 province con il maggior numero di casi positivi

Total_Positive_Cases_province = province_clean.groupby('ProvinceName')['Total_Positive_Cases_province'].sort_values(ascending=False)
top_10_province = Total_Positive_Cases_province.head(10)
print("Le dieci province con il maggior tasso di pazienti positivi sono:")
print(top_10_province)

# Produzione del grafico con colori diversi per ciascuna provincia
province = top_10_province.index
total_cases = top_10_province.values
plt.figure(figsize=(10, 6))
colors = ['r', 'purple', 'm', 'brown', 'orange', 'y', 'g', 'c', 'b', 'k']
for i in range(len(province)):
    plt.bar(province[i], total_cases[i], color=colors[i], label=province[i])
plt.title("Top 10 Province per Casi Positivi")
plt.xlabel("Provincia")
plt.ylabel("Casi Positivi")
plt.xticks(rotation=45)
plt.grid(True)
plt.legend()
plt.show()
```

Le dieci province con il maggior tasso di pazienti positivi sono:

ProvinceName

Milano 158717

Napoli 102702

Roma 96985

Torino 94349

Varese 45944

Monza e della Brianza 44603

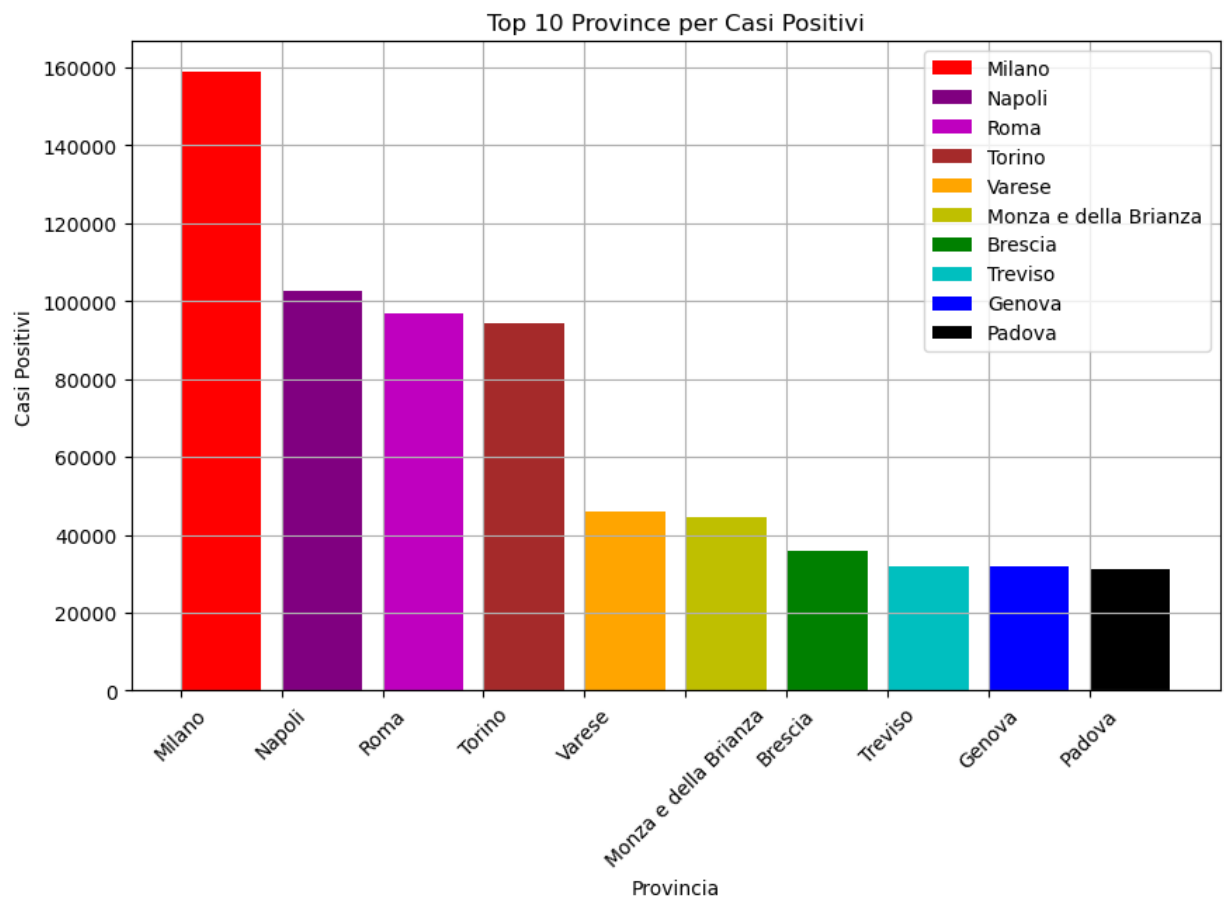
Brescia 36010

Treviso 31952

Genova 31864

Padova 31073

Name: TotalPositiveCases, dtype: int64





```
In [35]: # Leggi il file CSV dei poveri
poveri_regione = pd.read_csv('/Users/michelavenier/Desktop/EPICODE/BUILD_
poveri_regione = poveri_regione.drop(['Unnamed: 0'], axis=1)

# Converti la colonna Popolazione2011 in numerica
comuni_clean['Popolazione2011'] = pd.to_numeric(comuni_clean['Popolazione

# Calcola la somma della popolazione per regione
abitanti_regione = comuni_clean.groupby('Regione')['Popolazione2011'].sum
abitanti_regione = abitanti_regione.drop(0)

# Prendi il valore massimo di decessi per regione
regioni_raggruppate = regioni_clean.groupby('RegionName')['Deaths'].max()

# Unisci i dati dei poveri con i dati dei decessi
join_poveri_covidcase = pd.merge(poveri_regione, regioni_raggruppate, lef

# Unisci i dati completi
complessivo_contagi = pd.merge(join_poveri_covidcase, abitanti_regione, l

# Stampa i DataFrame
print("poveri_regione:", poveri_regione.shape, poveri_regione)
print("abitanti_regione:", abitanti_regione.shape, abitanti_regione)
print("regioni_raggruppate:", regioni_raggruppate.shape, regioni_raggrupp
print("join_poveri_covidcase:", join_poveri_covidcase.shape, join_poveri_
print("complessivo_contagi:", complesso_contagi.shape, complesso_conta
#problema con Valle d'Aosta

# Visualizzazione grafica
plt.figure(figsize=(15, 12))
plt.plot(complessivo_contagi['Territorio'], (complessivo_contagi['Deaths'
plt.bar(complessivo_contagi['Territorio'], complesso_contagi['Osservazi
plt.xticks(rotation=60)
plt.title('Essere povero ti fa morire?')
plt.ylabel('Morti ogni 100k abitanti vs. Indice povertà')
plt.legend()
plt.show()
```

```
poveri_regione: (20, 2)                                Territorio  Osservazione
0                Piemonte                                8.9
1            Valle d'Aosta                                6.3
2                Liguria                                10.6
3            Lombardia                                9.0
4  Trentino-Alto Adige/Sudtirolo                        4.8
5                Veneto                                8.0
6    Friuli-Venezia Giulia                              7.8
7            Emilia-Romagna                             7.6
8                Toscana                                7.8
9                Umbria                                10.8
10               Marche                                14.6
11               Lazio                                 6.4
12               Abruzzo                               15.5
13               Molise                                21.4
14               Campania                              26.3
15               Puglia                                23.2
```

16	Basilicata	27.0	
17	Calabria	24.1	
18	Sicilia	22.0	
19	Sardegna	16.6	
abitanti_regione: (20, 2)			Regione Popolazione20
11			
1	Abruzzo	1307309.0	
2	Basilicata	578036.0	
3	Calabria	1959050.0	
4	Campania	5766810.0	
5	Emilia-Romagna	4342135.0	
6	Friuli-Venezia Giulia	1220291.0	
7	Lazio	5502886.0	
8	Liguria	1570694.0	
9	Lombardia	9704151.0	
10	Marche	1541319.0	
11	Molise	313660.0	
12	Piemonte	4363916.0	
13	Puglia	4052566.0	
14	Sardegna	1639362.0	
15	Sicilia	5002904.0	
16	Toscana	3672202.0	
17	Trentino-Alto Adige/Sudtirolo	1029475.0	
18	Umbria	884268.0	
19	Valle d'Aosta	126806.0	
20	Veneto	4855904.0	
regioni_raggruppate: (20, 2)			RegionName Deaths
0	Abruzzo	980	
1	Basilicata	175	
2	Calabria	343	
3	Campania	1990	
4	Emilia-Romagna	6162	
5	Friuli-Venezia Giulia	1035	
6	Lazio	2622	
7	Liguria	2521	
8	Lombardia	23024	
9	Marche	1327	
10	Molise	135	
11	Piemonte	6623	
12	Puglia	1712	
13	Sardegna	531	
14	Sicilia	1759	
15	Toscana	2867	
16	Trentino-Alto Adige/Sudtirolo	691	
17	Umbria	460	
18	Valle d'Aosta	333	
19	Veneto	4210	
join_poveri_covidcase: (20, 4)			Territorio Osservaz
ione	RegionName \		
0	Piemonte	8.9	Piemo
n			
1	Valle d'Aosta	6.3	Valle d'Ao
2			
3	Liguria	10.6	Ligu
4			
5	Lombardia	9.0	Lombar
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

4	Trentino-Alto Adige/Sudtirolo	4.8	Trentino-Alto Adige/Sudti
5	Veneto	8.0	Ven
6	Friuli-Venezia Giulia	7.8	Friuli-Venezia Giu
7	Emilia-Romagna	7.6	Emilia-Roma
8	Toscana	7.8	Tosc
9	Umbria	10.8	Umb
10	Marche	14.6	Mar
11	Lazio	6.4	La
12	Abruzzo	15.5	Abru
13	Molise	21.4	Mol
14	Campania	26.3	Campa
15	Puglia	23.2	Pug
16	Basilicata	27.0	Basilic
17	Calabria	24.1	Calab
18	Sicilia	22.0	Sici
19	Sardegna	16.6	Sarde

	Deaths
0	6623
1	333
2	2521
3	23024
4	691
5	4210
6	1035
7	6162
8	2867
9	460
10	1327
11	2622
12	980
13	135
14	1990
15	1712
16	175
17	343
18	1759
19	531

complessivo_contagi: (20, 6)		Territorio	Osservazio
ne	RegionName \		
0	Piemonte	8.9	Piemo

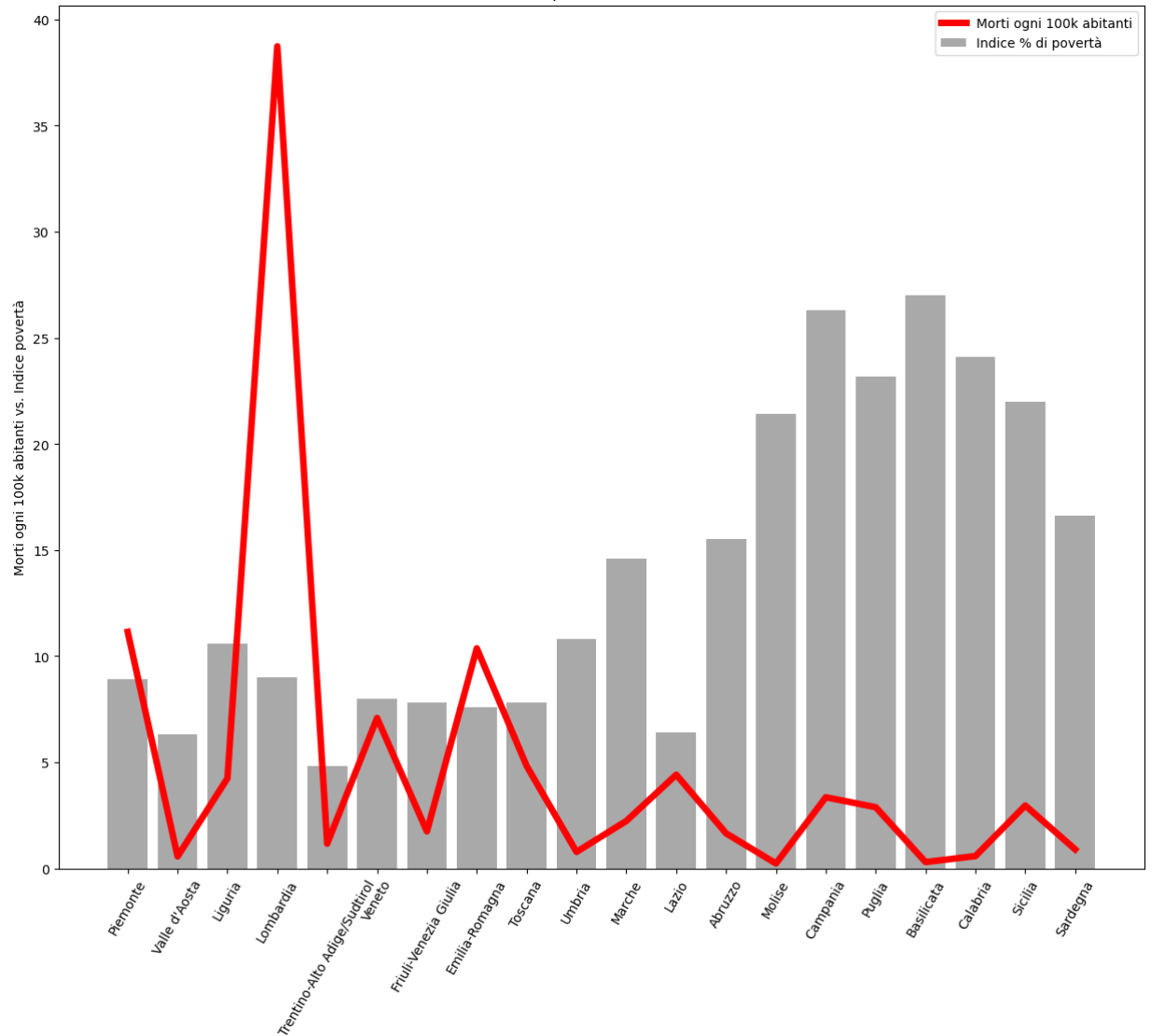
nte				
1	Valle d'Aosta	6.3	Valle d'Ao	
sta				
2	Liguria	10.6	Ligu	
ria				
3	Lombardia	9.0	Lombar	
dia				
4	Trentino-Alto Adige/Sudtirolo	4.8	Trentino-Alto Adige/Sudti	
rol				
5	Veneto	8.0	Ven	
eto				
6	Friuli-Venezia Giulia	7.8	Friuli-Venezia Giu	
lia				
7	Emilia-Romagna	7.6	Emilia-Roma	
gna				
8	Toscana	7.8	Tosc	
ana				
9	Umbria	10.8	Umb	
ria				
10	Marche	14.6	Mar	
che				
11	Lazio	6.4	La	
zio				
12	Abruzzo	15.5	Abru	
zzo				
13	Molise	21.4	Mol	
ise				
14	Campania	26.3	Campa	
nia				
15	Puglia	23.2	Pug	
lia				
16	Basilicata	27.0	Basilic	
ata				
17	Calabria	24.1	Calab	
ria				
18	Sicilia	22.0	Sici	
lia				
19	Sardegna	16.6	Sarde	
gna				

	Deaths	Regione	Popolazione2011
0	6623	Piemonte	4363916.0
1	333	Valle d'Aosta	126806.0
2	2521	Liguria	1570694.0
3	23024	Lombardia	9704151.0
4	691	Trentino-Alto Adige/Sudtirolo	1029475.0
5	4210	Veneto	4855904.0
6	1035	Friuli-Venezia Giulia	1220291.0
7	6162	Emilia-Romagna	4342135.0
8	2867	Toscana	3672202.0
9	460	Umbria	884268.0
10	1327	Marche	1541319.0
11	2622	Lazio	5502886.0
12	980	Abruzzo	1307309.0
13	135	Molise	313660.0
14	1990	Campania	5766810.0
15	1712	Puglia	4052566.0

16 175  
17 343  
18 1759  
19 531

Basilicata 578036.0  
Calabria 1959050.0  
Sicilia 5002904.0  
Sardegna 1639362.0

Essere povero ti fa morire?



```
In [36]: import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL del sito web
url = 'https://it.wikipedia.org/wiki/Regioni_d%27Italia#Dati_demografici_'

# Effettua una richiesta GET al sito
response = requests.get(url)

# Verifica se la richiesta è andata a buon fine
if response.status_code == 200:
    # Utilizza BeautifulSoup per analizzare il contenuto HTML della pagina
    soup = BeautifulSoup(response.content, 'html.parser')

    # Trova la tabella delle regioni
    table = soup.find('table', {'class': 'wikitable'})

    # Inizializza una lista per memorizzare i dati
    data = []

    # Estrai i dati dalla tabella
    rows = table.find_all('tr')
    for row in rows[1:]: # Salta la riga delle intestazioni
        cells = row.find_all(['th', 'td'])
        regione = cells[0].text.strip()
        superficie = cells[3].text.strip() # Assumendo che la superficie
        data.append([regione, superficie])

    # Crea un DataFrame pandas
    df = pd.DataFrame(data, columns=['Regione', 'Superficie_km2'])
    df = df.drop(df.index[-1])

    # Stampa il DataFrame
    print(df)
else:
    print('Errore nella richiesta HTTP')

print(ripartizioni_clean, regioni_clean['RegionName'].unique(), "df", df)
```

	Regione	Superficie_km2
0	Lombardia	23 863,10
1	Lazio	17 231,72
2	Campania	13 670,60
3	Veneto	18 345,37
4	Sicilia	25 832,55
5	Emilia-Romagna	22 501,43
6	Piemonte	25 386,70
7	Puglia	19 540,52
8	Toscana	22 987,44
9	Calabria	15 221,61
10	Sardegna	24 099,45
11	Liguria	5 416,15
12	Marche	9 344,29
13	Abruzzo	10 831,50
14	Friuli-Venezia Giulia	7 932,48

15	Trentino-Alto Adige	13	604,72
16	Umbria	8	464,22
17	Basilicata	10	073,11
18	Molise	4	460,44
19	Valle d'Aosta	3	260,85
	Codice Regione Ripartizione geografica		Regione
0	1	Nord-ovest	Piemonte
1	2	Nord-ovest	Valle d'Aosta
2	3	Nord-ovest	Lombardia
3	4	Nord-est	Trentino-Alto Adige/Sudtirolo
4	5	Nord-est	Veneto
5	6	Nord-est	Friuli-Venezia Giulia
6	7	Nord-ovest	Liguria
7	8	Nord-est	Emilia-Romagna
8	9	Centro	Toscana
9	10	Centro	Umbria
10	11	Centro	Marche
11	12	Centro	Lazio
12	13	Sud	Abruzzo
13	14	Sud	Molise
14	15	Sud	Campania
15	16	Sud	Puglia
16	17	Sud	Basilicata
17	18	Sud	Calabria
18	19	Isole	Sicilia
19	20	Isole	Sardegna

```
[ 'Abruzzo' 'Basilicata' 'Calabria' 'Campania' 'Emilia-Romagna'
  'Friuli-Venezia Giulia' 'Lazio' 'Liguria' 'Lombardia' 'Marche' 'Molise'
  'Trentino-Alto Adige/Sudtirolo' 'Piemonte' 'Puglia' 'Sardegna' 'Sicilia'
  'Toscana' 'Umbria' "Valle d'Aosta" 'Veneto'] df
e Superficie_km2
```

0	Lombardia	23	863,10
1	Lazio	17	231,72
2	Campania	13	670,60
3	Veneto	18	345,37
4	Sicilia	25	832,55
5	Emilia-Romagna	22	501,43
6	Piemonte	25	386,70
7	Puglia	19	540,52
8	Toscana	22	987,44
9	Calabria	15	221,61
10	Sardegna	24	099,45
11	Liguria	5	416,15
12	Marche	9	344,29
13	Abruzzo	10	831,50
14	Friuli-Venezia Giulia	7	932,48
15	Trentino-Alto Adige	13	604,72
16	Umbria	8	464,22
17	Basilicata	10	073,11
18	Molise	4	460,44
19	Valle d'Aosta	3	260,85

```
In [37]: import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL del sito web
```

```

url = 'https://it.wikipedia.org/wiki/Regioni_d%27Italia#Dati_demografici_

# Effettua una richiesta GET al sito
response = requests.get(url)

# Verifica se la richiesta è andata a buon fine
if response.status_code == 200:
    # Utilizza BeautifulSoup per analizzare il contenuto HTML della pagina
    soup = BeautifulSoup(response.content, 'html.parser')

    # Trova la tabella delle regioni
    table = soup.find('table', {'class': 'wikitable'})

    # Inizializza una lista per memorizzare i dati
    data = []

    # Estrai i dati dalla tabella
    rows = table.find_all('tr')
    for row in rows[1:]: # Salta la riga delle intestazioni
        cells = row.find_all(['th', 'td'])
        regione = cells[0].text.strip()
        superficie = cells[3].text.strip() # Assumendo che la superficie
        data.append([regione, superficie])

    # Crea un DataFrame pandas
    df = pd.DataFrame(data, columns=['Regione', 'Superficie_km2'])
    df = df.drop(df.index[-1])

    # Stampa il DataFrame
    print(df)
else:
    print('Errore nella richiesta HTTP')

print(ripartizioni_clean, regioni_clean['RegionName'].unique(), "df", df)
df['Superficie_km2'] = df['Superficie_km2'].replace('trentino-alto adige',

abitanti_regione=comuni_clean.groupby('Regione')['Popolazione2011'].sum()
abitanti_regione['Regione']=abitanti_regione['Regione'].str.lower()
abitanti_regione['Regione'] = abitanti_regione['Regione'].replace('trenti

df['Regione']=df['Regione'].str.lower()
df['Superficie_km2'] = df['Superficie_km2'].str.replace(',', '.').str.rep
df['Superficie_km2'] = df['Superficie_km2'].replace('trentino-alto adige',

calcolo_densita=pd.merge(abitanti_regione, df, left_on='Regione', right_o
calcolo_densita['Superficie_km2']=calcolo_densita['Superficie_km2'].astyp
calcolo_densita['densita']=calcolo_densita['Popolazione2011']/calcolo_den
regioni_raggruppate['RegionName']=regioni_raggruppate['RegionName'].str.l

morti_densita=pd.merge(calcolo_densita, regioni_raggruppate, left_on='Reg
print("morti_densita", morti_densita)

plt.figure(figsize=(15,12))
plt.bar(morti_densita['Regione'],morti_densita['densita']*50, color='c',
plt.plot(morti_densita['Regione'],morti_densita['Deaths'], color='r',line

```



```
plt.xticks(rotation=60)
plt.title('Come influisce la densità di popolazione sul contagio?')
plt.ylabel('Morti vs densità')
plt.legend()
plt.show()
```

	Regione	Superficie_km2
0	Lombardia	23 863,10
1	Lazio	17 231,72
2	Campania	13 670,60
3	Veneto	18 345,37
4	Sicilia	25 832,55
5	Emilia-Romagna	22 501,43
6	Piemonte	25 386,70
7	Puglia	19 540,52
8	Toscana	22 987,44
9	Calabria	15 221,61
10	Sardegna	24 099,45
11	Liguria	5 416,15
12	Marche	9 344,29
13	Abruzzo	10 831,50
14	Friuli-Venezia Giulia	7 932,48
15	Trentino-Alto Adige	13 604,72
16	Umbria	8 464,22
17	Basilicata	10 073,11
18	Molise	4 460,44
19	Valle d'Aosta	3 260,85

	Codice	Regione	Ripartizione geografica	Regione
0	1		Nord-ovest	Piemonte
1	2		Nord-ovest	Valle d'Aosta
2	3		Nord-ovest	Lombardia
3	4		Nord-est	Trentino-Alto Adige/Sudtirolo
4	5		Nord-est	Veneto
5	6		Nord-est	Friuli-Venezia Giulia
6	7		Nord-ovest	Liguria
7	8		Nord-est	Emilia-Romagna
8	9		Centro	Toscana
9	10		Centro	Umbria
10	11		Centro	Marche
11	12		Centro	Lazio
12	13		Sud	Abruzzo
13	14		Sud	Molise
14	15		Sud	Campania
15	16		Sud	Puglia
16	17		Sud	Basilicata
17	18		Sud	Calabria
18	19		Isole	Sicilia
19	20		Isole	Sardegna

```
[ 'Abruzzo' 'Basilicata' 'Calabria' 'Campania' 'Emilia-Romagna'
  'Friuli-Venezia Giulia' 'Lazio' 'Liguria' 'Lombardia' 'Marche' 'Molise'
  'Trentino-Alto Adige/Sudtirolo' 'Piemonte' 'Puglia' 'Sardegna' 'Sicilia'
  'Toscana' 'Umbria' "Valle d'Aosta" 'Veneto' ] df
e Superficie_km2
```

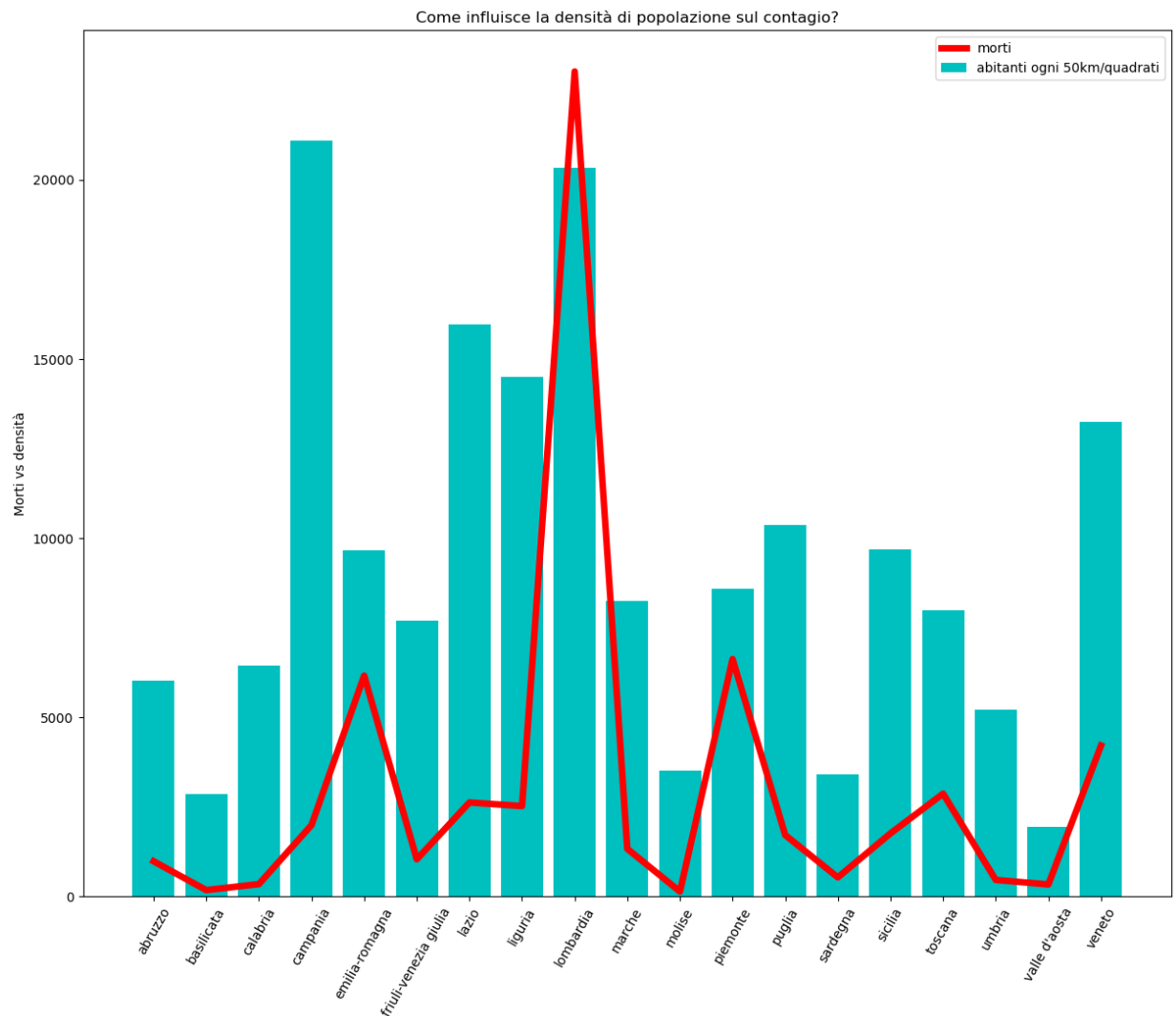
	Regione	Superficie_km2
0	Lombardia	23 863,10
1	Lazio	17 231,72
2	Campania	13 670,60
3	Veneto	18 345,37

4	Sicilia	25	832,55	
5	Emilia-Romagna	22	501,43	
6	Piemonte	25	386,70	
7	Puglia	19	540,52	
8	Toscana	22	987,44	
9	Calabria	15	221,61	
10	Sardegna	24	099,45	
11	Liguria	5	416,15	
12	Marche	9	344,29	
13	Abruzzo	10	831,50	
14	Friuli-Venezia Giulia	7	932,48	
15	Trentino-Alto Adige	13	604,72	
16	Umbria	8	464,22	
17	Basilicata	10	073,11	
18	Molise	4	460,44	
19	Valle d'Aosta	3	260,85	
morti_densita		Regione	Popolazione2011	Superficie_km2
densita \				
0	abruzzo	1307309.0	10831.50	120.695102
1	basilicata	578036.0	10073.11	57.384065
2	calabria	1959050.0	15221.61	128.701892
3	campania	5766810.0	13670.60	421.840300
4	emilia-romagna	4342135.0	22501.43	192.971513
5	friuli-venezia giulia	1220291.0	7932.48	153.834740
6	lazio	5502886.0	17231.72	319.346299
7	liguria	1570694.0	5416.15	290.001939
8	lombardia	9704151.0	23863.10	406.659277
9	marche	1541319.0	9344.29	164.947685
10	molise	313660.0	4460.44	70.320417
11	piemonte	4363916.0	25386.70	171.897726
12	puglia	4052566.0	19540.52	207.392946
13	sardegna	1639362.0	24099.45	68.024872
14	sicilia	5002904.0	25832.55	193.666672
15	toscana	3672202.0	22987.44	159.748193
16	umbria	884268.0	8464.22	104.471292
17	valle d'aosta	126806.0	3260.85	38.887407
18	veneto	4855904.0	18345.37	264.693707

	RegionName	Deaths
0	abruzzo	980
1	basilicata	175
2	calabria	343
3	campania	1990
4	emilia-romagna	6162
5	friuli-venezia giulia	1035
6	lazio	2622
7	liguria	2521
8	lombardia	23024
9	marche	1327
10	molise	135
11	piemonte	6623
12	puglia	1712
13	sardegna	531
14	sicilia	1759
15	toscana	2867
16	umbria	460
17	valle d'aosta	333

18

veneto 4210



In [38]: *# box plot che indica la distribuzione dei Decessi per Covid-19 in Italia*

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

# i dati della regione Lombardia, molto più grandi rispetto a quelli dell
# Se si vogliono visualizzare i box plot delle altre regioni in maggiore
#regioni_decessi = regioni_clean[regioni_clean['RegionName'] != 'Lombardi
#regioni_decessi = regioni_clean['RegionName']

regioni_somma_decessi = regioni_clean.groupby('RegionName')['Deaths']
print(regioni_somma_decessi.describe())

# Plot del boxplot
plt.figure(figsize=(10, 6))
sns.boxplot(data=regioni_clean, x='RegionName', y='Deaths')
plt.xlabel('Regione')
plt.ylabel('Picco dei decessi')
plt.title('Box Plot Decessi Covid in Italia per Regione')
plt.xticks(rotation=75) # Rotazione delle etichette sull'asse x per una
plt.show()
```

count

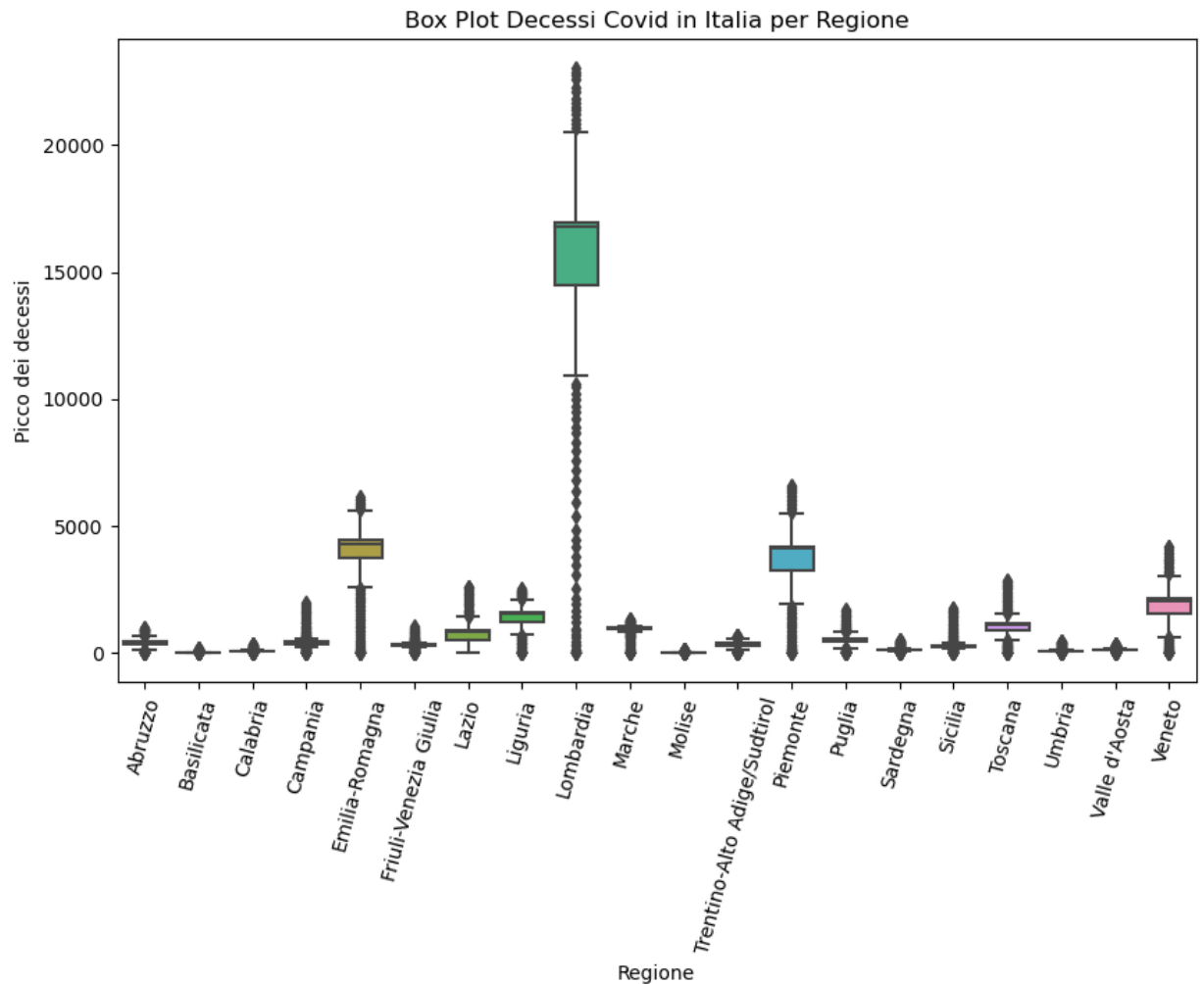
mean

std min

25% \					
RegionName					
Abruzzo	287.0	412.188153	205.593205	0.0	33
8.0					
Basilicata	287.0	33.801394	31.523090	0.0	2
5.0					
Calabria	287.0	97.013937	58.899165	0.0	8
8.5					
Campania	287.0	470.588850	351.108681	0.0	37
2.5					
Emilia-Romagna	287.0	3758.376307	1454.794924	0.0	372
1.0					
Friuli-Venezia Giulia	287.0	325.515679	173.187635	0.0	30
4.5					
Lazio	287.0	811.693380	532.126717	0.0	53
6.0					
Liguria	287.0	1357.435540	596.388034	0.0	123
7.5					
Lombardia	287.0	14552.163763	5457.487022	6.0	1450
0.0					
Marche	287.0	868.832753	318.123794	0.0	93
9.5					
Molise	287.0	27.383275	24.659630	0.0	2
2.0					
Piemonte	287.0	3491.442509	1563.696390	0.0	323
1.5					
Puglia	287.0	529.808362	317.710827	0.0	43
5.5					
Sardegna	287.0	143.620209	99.586235	0.0	11
9.0					
Sicilia	287.0	345.296167	328.442318	0.0	24
8.5					
Toscana	287.0	1038.975610	571.448506	0.0	89
4.0					
Trentino-Alto Adige/Sudtirolo	574.0	321.630662	146.897001	0.0	29
1.0					
Umbria	287.0	95.536585	87.169343	0.0	7
0.0					
Valle d'Aosta	287.0	138.937282	67.279057	0.0	13
9.0					
Veneto	287.0	1819.912892	878.388585	1.0	155
6.5					

	50%	75%	max
RegionName			
Abruzzo	468.0	477.0	980.0
Basilicata	27.0	28.0	175.0
Calabria	97.0	99.0	343.0
Campania	432.0	459.5	1990.0
Emilia-Romagna	4271.0	4481.0	6162.0
Friuli-Venezia Giulia	345.0	350.0	1035.0
Lazio	848.0	905.0	2622.0
Liguria	1564.0	1598.0	2521.0
Lombardia	16775.0	16939.0	23024.0
Marche	987.0	992.0	1327.0
Molise	23.0	23.0	135.0
Piemonte	4118.0	4159.5	6623.0

Puglia	548.0	588.5	1712.0
Sardegna	134.0	148.0	531.0
Sicilia	283.0	306.0	1759.0
Toscana	1127.0	1156.5	2867.0
Trentino-Alto Adige/Sudtirolo	293.5	406.0	691.0
Umbria	80.0	85.0	460.0
Valle d'Aosta	146.0	146.0	333.0
Veneto	2047.0	2173.5	4210.0



```
In [39]: # aggiungo la colonna ripartizioni geografiche alle regioni facendo un join
# questo si potrebbe utilizzare per analisi ulteriori sull'analisi dell'e

import pandas as pd

result = pd.merge(regioni_clean, ripartizioni_clean, left_on='RegionCode'
print(result.head(2))
```

	SNo	Country	RegionCode	RegionName	Latitude	Longitude	\
0	0	ITA	13	Abruzzo	42.351222	13.398438	
1	21	ITA	13	Abruzzo	42.351222	13.398438	

	HospitalizedPatients	IntensiveCarePatients	TotalHospitalizedPatients
\			
0	0	0	0
1	0	0	0

	HomeConfinement	CurrentPositiveCases	NewPositiveCases	Recovered	Deaths
\					
0	0	0	0	0	0
0					
1	0	0	0	0	0
0					

	TotalPositiveCases	TestsPerformed	Codice Regione	Ripartizione geografica
\				
0	0	NaN	13	
Sud				
1	0	NaN	13	
Sud				

	Regione
0	Abruzzo
1	Abruzzo