# m4-es-4-2-esamefinale-vb

April 9, 2024

[ ]:
```
# Importo il datatset denominato owid-covid-data in csv e verifico le prime 10 righe del dataframe
```

[91]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("owid-covid-data.csv")
```

[93]:
```
df.head()
```

[93]:

| | iso_code | continent | location | date | total_cases | new_cases | \ |
|---|---|---|---|---|---|---|---|
| 0 | AFG | Asia | Afghanistan | 2020-01-05 | NaN | 0.0 | |
| 1 | AFG | Asia | Afghanistan | 2020-01-06 | NaN | 0.0 | |
| 2 | AFG | Asia | Afghanistan | 2020-01-07 | NaN | 0.0 | |
| 3 | AFG | Asia | Afghanistan | 2020-01-08 | NaN | 0.0 | |
| 4 | AFG | Asia | Afghanistan | 2020-01-09 | NaN | 0.0 | |

| | new_cases_smoothed | total_deaths | new_deaths | new_deaths_smoothed | … | \ |
|---|---|---|---|---|---|---|
| 0 | NaN | NaN | 0.0 | NaN | … | |
| 1 | NaN | NaN | 0.0 | NaN | … | |
| 2 | NaN | NaN | 0.0 | NaN | … | |
| 3 | NaN | NaN | 0.0 | NaN | … | |
| 4 | NaN | NaN | 0.0 | NaN | … | |

| | male_smokers | handwashing_facilities | hospital_beds_per_thousand | \ |
|---|---|---|---|---|
| 0 | NaN | 37.746 | 0.5 | |
| 1 | NaN | 37.746 | 0.5 | |
| 2 | NaN | 37.746 | 0.5 | |
| 3 | NaN | 37.746 | 0.5 | |
| 4 | NaN | 37.746 | 0.5 | |

| | life_expectancy | human_development_index | population | \ |
|---|---|---|---|---|
| 0 | 64.83 | 0.511 | 41128772.0 | |
| 1 | 64.83 | 0.511 | 41128772.0 | |
| 2 | 64.83 | 0.511 | 41128772.0 | |
| 3 | 64.83 | 0.511 | 41128772.0 | |

```
4            64.83                    0.511  41128772.0

   excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
0                                   NaN                          NaN
1                                   NaN                          NaN
2                                   NaN                          NaN
3                                   NaN                          NaN
4                                   NaN                          NaN

   excess_mortality  excess_mortality_cumulative_per_million
0               NaN                                      NaN
1               NaN                                      NaN
2               NaN                                      NaN
3               NaN                                      NaN
4               NaN                                      NaN

[5 rows x 67 columns]
```

## 1 Verifico le dimensioni del dataframe

[6]:
```
df.shape
```

[6]:
```
(384091, 67)
```

## 2 Verifico le diciture presenti nell'intestazione

[94]:
```
df.columns.values.tolist()
```

[94]:
```
['iso_code',
 'continent',
 'location',
 'date',
 'total_cases',
 'new_cases',
 'new_cases_smoothed',
 'total_deaths',
 'new_deaths',
 'new_deaths_smoothed',
 'total_cases_per_million',
 'new_cases_per_million',
 'new_cases_smoothed_per_million',
 'total_deaths_per_million',
 'new_deaths_per_million',
 'new_deaths_smoothed_per_million',
 'reproduction_rate',
```

```
'icu_patients',
'icu_patients_per_million',
'hosp_patients',
'hosp_patients_per_million',
'weekly_icu_admissions',
'weekly_icu_admissions_per_million',
'weekly_hosp_admissions',
'weekly_hosp_admissions_per_million',
'total_tests',
'new_tests',
'total_tests_per_thousand',
'new_tests_per_thousand',
'new_tests_smoothed',
'new_tests_smoothed_per_thousand',
'positive_rate',
'tests_per_case',
'tests_units',
'total_vaccinations',
'people_vaccinated',
'people_fully_vaccinated',
'total_boosters',
'new_vaccinations',
'new_vaccinations_smoothed',
'total_vaccinations_per_hundred',
'people_vaccinated_per_hundred',
'people_fully_vaccinated_per_hundred',
'total_boosters_per_hundred',
'new_vaccinations_smoothed_per_million',
'new_people_vaccinated_smoothed',
'new_people_vaccinated_smoothed_per_hundred',
'stringency_index',
'population_density',
'median_age',
'aged_65_older',
'aged_70_older',
'gdp_per_capita',
'extreme_poverty',
'cardiovasc_death_rate',
'diabetes_prevalence',
'female_smokers',
'male_smokers',
'handwashing_facilities',
'hospital_beds_per_thousand',
'life_expectancy',
'human_development_index',
'population',
'excess_mortality_cumulative_absolute',
```

```
 'excess_mortality_cumulative',
 'excess_mortality',
 'excess_mortality_cumulative_per_million']
```

## 3 Verifichiamo la tipologia dei dati

[95]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 384091 entries, 0 to 384090
Data columns (total 67 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   iso_code                                384091 non-null  object
 1   continent                               365633 non-null  object
 2   location                                384091 non-null  object
 3   date                                    384091 non-null  object
 4   total_cases                             344917 non-null  float64
 5   new_cases                               372892 non-null  float64
 6   new_cases_smoothed                      371662 non-null  float64
 7   total_deaths                            322777 non-null  float64
 8   new_deaths                              373187 non-null  float64
 9   new_deaths_smoothed                     371957 non-null  float64
 10  total_cases_per_million                 344917 non-null  float64
 11  new_cases_per_million                   372892 non-null  float64
 12  new_cases_smoothed_per_million          371662 non-null  float64
 13  total_deaths_per_million                322777 non-null  float64
 14  new_deaths_per_million                  373187 non-null  float64
 15  new_deaths_smoothed_per_million         371957 non-null  float64
 16  reproduction_rate                       184817 non-null  float64
 17  icu_patients                            38625 non-null   float64
 18  icu_patients_per_million                38625 non-null   float64
 19  hosp_patients                           40158 non-null   float64
 20  hosp_patients_per_million               40158 non-null   float64
 21  weekly_icu_admissions                   10674 non-null   float64
 22  weekly_icu_admissions_per_million       10674 non-null   float64
 23  weekly_hosp_admissions                  24149 non-null   float64
 24  weekly_hosp_admissions_per_million      24149 non-null   float64
 25  total_tests                             79387 non-null   float64
 26  new_tests                               75403 non-null   float64
 27  total_tests_per_thousand                79387 non-null   float64
 28  new_tests_per_thousand                  75403 non-null   float64
 29  new_tests_smoothed                      103965 non-null  float64
 30  new_tests_smoothed_per_thousand         103965 non-null  float64
 31  positive_rate                           95927 non-null   float64
 32  tests_per_case                          94348 non-null   float64
```

```
 33  tests_units                                  106788 non-null  object
 34  total_vaccinations                           83270 non-null   float64
 35  people_vaccinated                            79152 non-null   float64
 36  people_fully_vaccinated                      76032 non-null   float64
 37  total_boosters                               51459 non-null   float64
 38  new_vaccinations                             69014 non-null   float64
 39  new_vaccinations_smoothed                    189974 non-null  float64
 40  total_vaccinations_per_hundred               83270 non-null   float64
 41  people_vaccinated_per_hundred                79152 non-null   float64
 42  people_fully_vaccinated_per_hundred          76032 non-null   float64
 43  total_boosters_per_hundred                   51459 non-null   float64
 44  new_vaccinations_smoothed_per_million        189974 non-null  float64
 45  new_people_vaccinated_smoothed               187348 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred   187348 non-null  float64
 47  stringency_index                             197292 non-null  float64
 48  population_density                           326464 non-null  float64
 49  median_age                                   303493 non-null  float64
 50  aged_65_older                                293024 non-null  float64
 51  aged_70_older                                300453 non-null  float64
 52  gdp_per_capita                               297565 non-null  float64
 53  extreme_poverty                              191823 non-null  float64
 54  cardiovasc_death_rate                        298135 non-null  float64
 55  diabetes_prevalence                          313509 non-null  float64
 56  female_smokers                               223827 non-null  float64
 57  male_smokers                                 220787 non-null  float64
 58  handwashing_facilities                       146018 non-null  float64
 59  hospital_beds_per_thousand                   263347 non-null  float64
 60  life_expectancy                              353653 non-null  float64
 61  human_development_index                      289180 non-null  float64
 62  population                                   384091 non-null  float64
 63  excess_mortality_cumulative_absolute         13172 non-null   float64
 64  excess_mortality_cumulative                  13172 non-null   float64
 65  excess_mortality                             13172 non-null   float64
 66  excess_mortality_cumulative_per_million      13172 non-null   float64
dtypes: float64(62), object(5)
memory usage: 196.3+ MB
```

### 3.1 Determino i casi totali per continente

[96]:
```
continent_totalcases=df.groupby("continent")["new_cases"].sum()
```

[97]:
```
continent_totalcases
```

[97]:
```
continent
Africa            13140432.0
Asia             301426766.0
Europe           252322670.0
```

```
North America    124525279.0
Oceania           14791186.0
South America     68695341.0
Name: new_cases, dtype: float64
```

### 3.2 Prendo 2 continenti e confronto i seguenti descrittori statistici: valori minimo e massimo, media, e percentuale rispetto al numero dei casi totali nel mondo

[98]:
```
europe=df[df["continent"]=="Europe"]
```

[99]:
```
europe_tot=europe.groupby(["continent"])["new_cases"].sum()
print(europe_tot)
```

```
continent
Europe    252322670.0
Name: new_cases, dtype: float64
```

[101]:
```
europe_perc=europe_tot/ continent_totalcases.sum()
print(europe_perc)
```

```
continent
Europe    0.325619
Name: new_cases, dtype: float64
```

[102]:
```
europe_data=europe.groupby(["continent"])["new_cases"].
↪agg(["sum","max","min","mean"]).round()
print(europe_data)
```

```
                  sum        max  min    mean
continent
Europe     252322670.0  2417043.0  0.0  3286.0
```

### 3.3 Unisco i dati statistici e la percentuale in un uncia tabella denominata eu

[103]:
```
eu=pd.merge(europe_perc,europe_data,on="continent")
print(eu)
```

```
           new_cases          sum        max  min    mean
continent
Europe      0.325619  252322670.0  2417043.0  0.0  3286.0
```

### 3.4 Eseguo le stesse operazioni per l'oceania

[104]:
```
oceania=df[df["continent"]=="Oceania"]
```

[105]:
```
oceania_tot=oceania.groupby(["continent"])["new_cases"].sum()
print(oceania_tot)
```

```
continent
Oceania    14791186.0
Name: new_cases, dtype: float64
```

[106]:
```
oceania_perc=oceania_tot/ continent_totalcases.sum()
print(oceania_perc)
```

```
continent
Oceania    0.019088
Name: new_cases, dtype: float64
```

[107]:
```
oceania_data=oceania.groupby(["continent"])["new_cases"].
  ↪agg(["sum","max","min","mean"]).round()
print(oceania_data)
```

```
                 sum       max  min   mean
continent
Oceania    14791186.0  588813.0  0.0  405.0
```

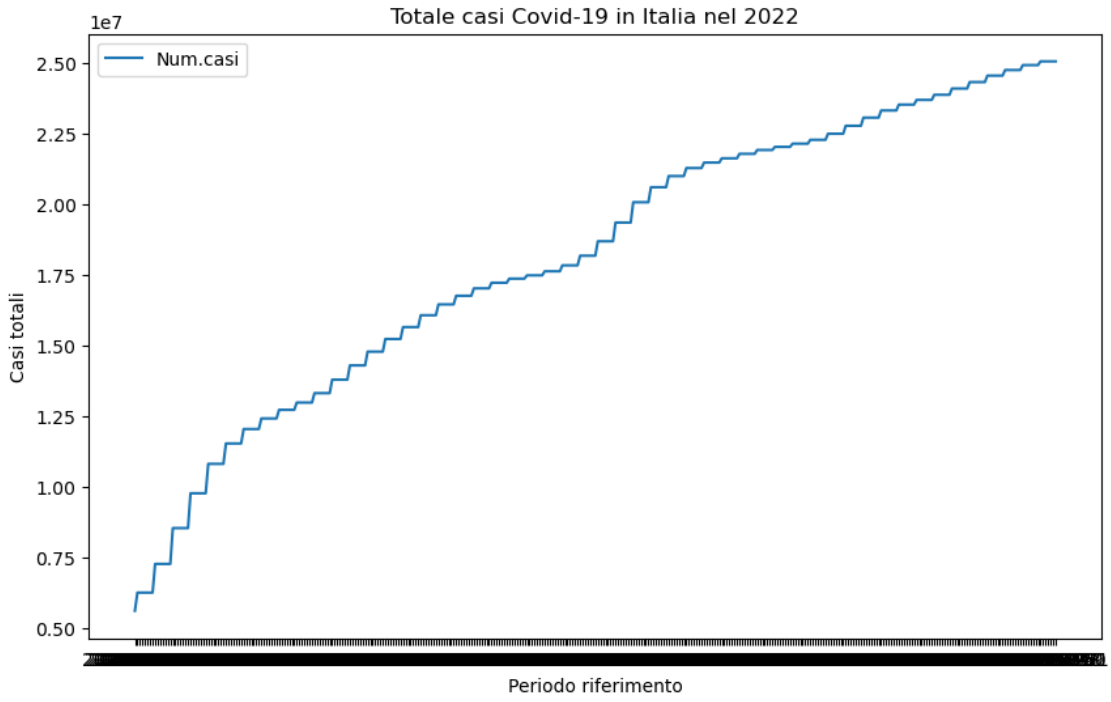
[63]:
```
oc=pd.merge(oceania_perc,oceania_data,on="continent")
print(oc)
```

```
           new_cases         sum       max  min   mean
continent
Oceania     0.019088  14791186.0  588813.0  0.0  405.0
```

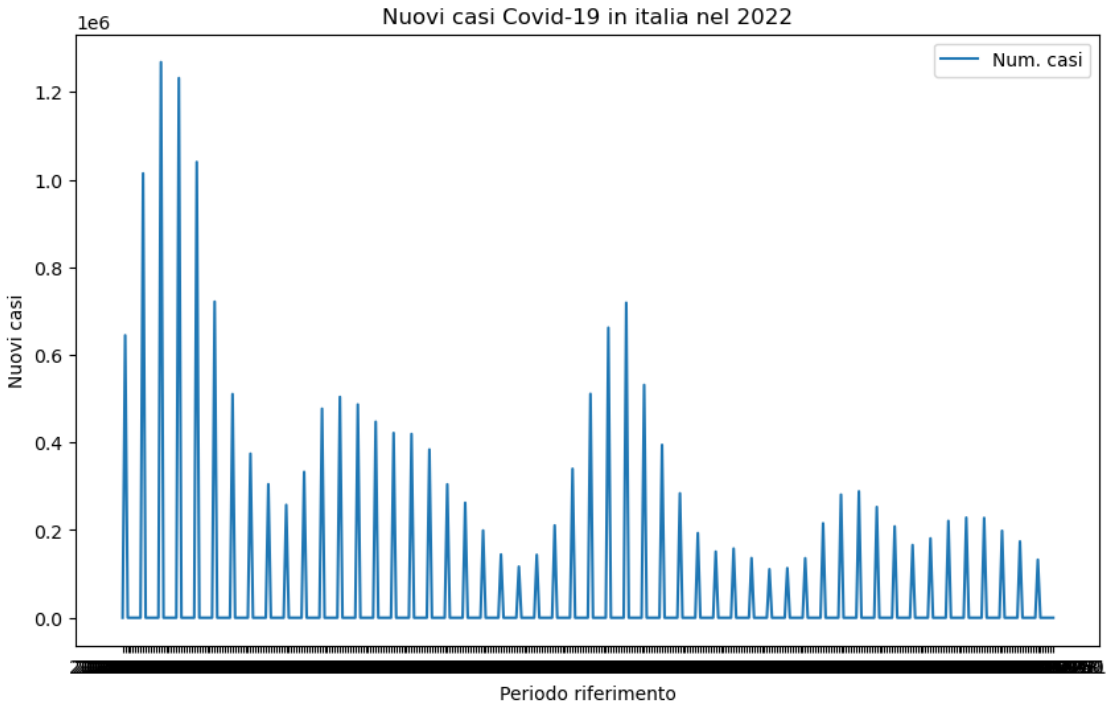
### 3.5 Prendo in esame i dati covid relativi all'Italia nel 2022, e mostro con un grafico: 1)l'evoluzione dei casi totali, 2) il numero di nuovi casi rispetto alla data 3) l'andamento della somma cumulativa dei nuovi casi del 2022

[111]:
```
italia_22=df[(df["location"]=="Italy")&(df["date"].str.startswith("2022"))]
```

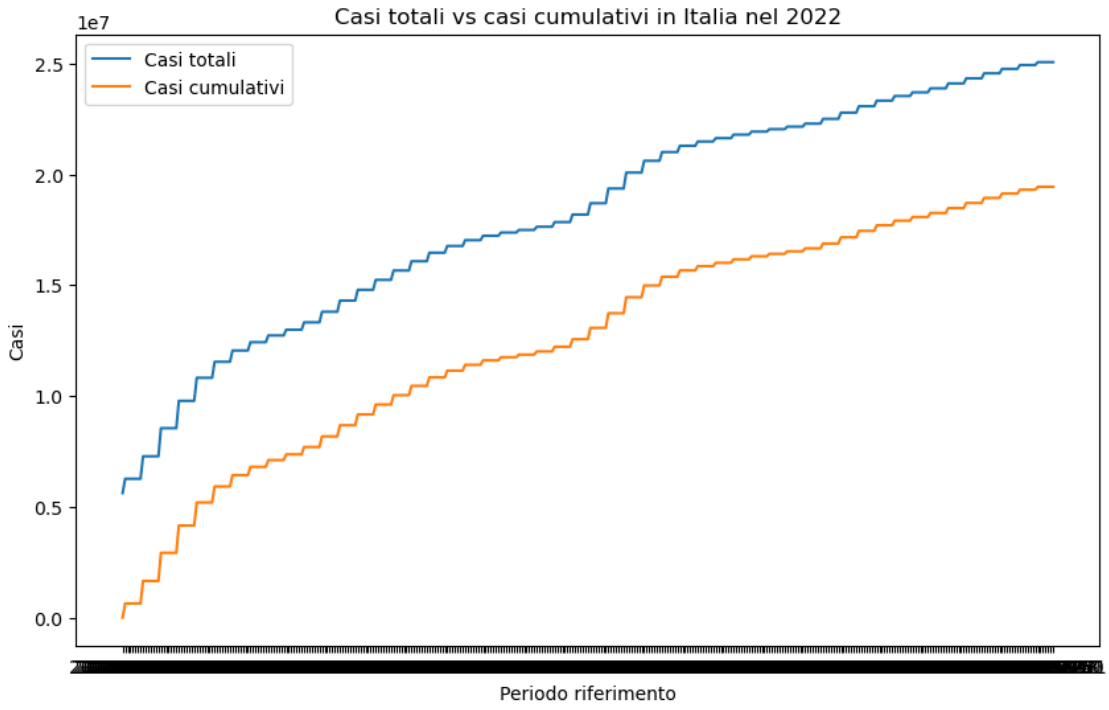
[114]:
```
plt.figure(figsize=(10, 6))
plt.plot(italia_22["date"], italia_22["total_cases"], label="Num.casi")
plt.xlabel("Periodo riferimento")
plt.ylabel("Casi totali")
plt.title("Totale casi Covid-19 in Italia nel 2022")
plt.legend()
plt.show()
```




[115]:
```
plt.figure(figsize=(10, 6))
plt.plot(italia_22["date"], italia_22["new_cases"], label="Num. casi")
plt.xlabel("Periodo riferimento")
plt.ylabel("Nuovi casi")
plt.title("Nuovi casi Covid-19 in italia nel 2022")
plt.legend()
plt.show()
```




[116]:
```
ita_andamento=italia_22["new_cases"].cumsum()
plt.figure(figsize=(10, 6))
plt.plot(italia_22["date"], italia_22["total_cases"], label="Casi totali")
plt.plot(italia_22["date"], ita_andamento, label="Casi cumulativi")
plt.xlabel("Periodo riferimento")
plt.ylabel("Casi")
plt.title("Casi totali vs casi cumulativi in Italia nel 2022")
plt.legend()
plt.show()
```




**3.6 I casi totali ed i casi cumulativi seguono lo stesso andamento ed entrambe mostrano un picco intorno alla fine dell'anno (nov-dic) per poi tornare ad essere costante.**

**3.7 Filtro il dataframe per gli stati Italia, Germania e Francia, e mostro in un boxplot la differenza tra queste nazioni riguardo il numero di pazienti in terapia intensiva (Intensive Care Unit, ICU) da maggio 2022 (incluso) ad aprile 2023 (incluso)**

[154]:
```
igf_data_mediana=igf_data.groupby(["location"])["icu_patients"].median().round()
igf_data_mediana
```

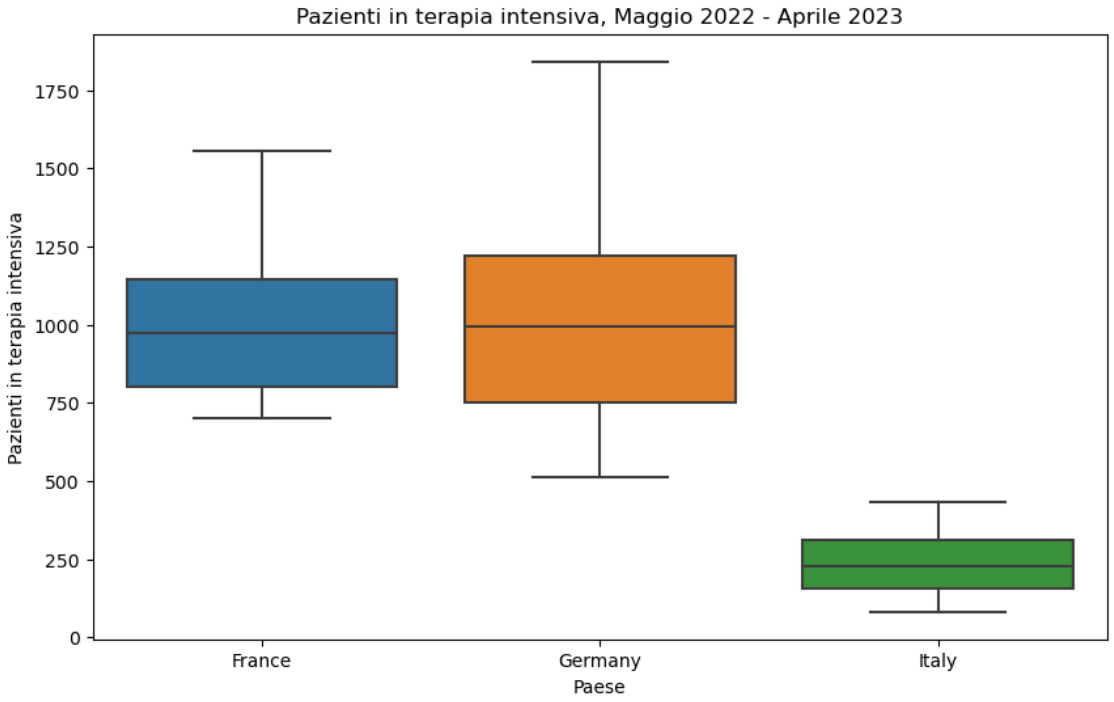
[154]:
```
location
France     972.0
Germany    994.0
Italy      227.0
Name: icu_patients, dtype: float64
```

[153]:
```
igf_data_media=igf_data.groupby(["location"])["icu_patients"].mean().round()
igf_data_media
```

[153]:
```
location
France      998.0
Germany    1022.0
```

```
Italy       231.0
Name: icu_patients, dtype: float64
```

[151]:
```
igf_data=df[(df["location"].isin(["Italy", "Germany", "France"])) & (df["date"].
  ↪between("2022-05-01", "2023-04-30"))]
plt.figure(figsize=(10, 6))
sns.boxplot(x="location", y='icu_patients', data=igf_data)
plt.title("Pazienti in terapia intensiva, Maggio 2022 - Aprile 2023")
plt.xlabel("Paese")
plt.ylabel("Pazienti in terapia intensiva")
plt.show()
```



**3.8 I boxplot dimostrano che Francia e Germania hanno avuto un valore mediano di pazienti in terapia intensiva più alto rispetto all'Italia, ma quest'ultima è la nazione con una distribuzione perfettamente simmetrica, infatti la mediana si colloca esattamente in posizione centrale rispetto alla scatola, ovvero alla stessa distanza tra primo e terzo quartile e la media (231) coincide con la mediana (227). La Francia e la Germania presentano una asimmetria, cioè una tendenza dei dati a disperdersi verso valori più grandi rispetto a quello centrale.**

[139]:
```
igfs_data=df[(df["location"].isin(["Italy", "Germany", "France","Spain"])) &␣
 ↪(df["date"].between("2023-01-01", "2023-12-31"))]
igfs_data.head()
```

[139]:

| | iso_code | continent | location | date | total_cases | new_cases | \ |
|---|---|---|---|---|---|---|---|
| 116679 | FRA | Europe | France | 2023-01-01 | 38141254.0 | 151707.0 | |
| 116680 | FRA | Europe | France | 2023-01-02 | 38141254.0 | 0.0 | |
| 116681 | FRA | Europe | France | 2023-01-03 | 38141254.0 | 0.0 | |
| 116682 | FRA | Europe | France | 2023-01-04 | 38141254.0 | 0.0 | |
| 116683 | FRA | Europe | France | 2023-01-05 | 38141254.0 | 0.0 | |

| | new_cases_smoothed | total_deaths | new_deaths | new_deaths_smoothed | \ |
|---|---|---|---|---|---|
| 116679 | 21672.429 | 162475.0 | 808.0 | 115.429 | |
| 116680 | 21672.429 | 162475.0 | 0.0 | 115.429 | |
| 116681 | 21672.429 | 162475.0 | 0.0 | 115.429 | |
| 116682 | 21672.429 | 162475.0 | 0.0 | 115.429 | |
| 116683 | 21672.429 | 162475.0 | 0.0 | 115.429 | |

| | … | male_smokers | handwashing_facilities | hospital_beds_per_thousand | \ |
|---|---|---|---|---|---|
| 116679 | … | 35.6 | NaN | 5.98 | |
| 116680 | … | 35.6 | NaN | 5.98 | |
| 116681 | … | 35.6 | NaN | 5.98 | |
| 116682 | … | 35.6 | NaN | 5.98 | |
| 116683 | … | 35.6 | NaN | 5.98 | |

| | life_expectancy | human_development_index | population | \ |
|---|---|---|---|---|
| 116679 | 82.66 | 0.901 | 67813000.0 | |
| 116680 | 82.66 | 0.901 | 67813000.0 | |
| 116681 | 82.66 | 0.901 | 67813000.0 | |
| 116682 | 82.66 | 0.901 | 67813000.0 | |
| 116683 | 82.66 | 0.901 | 67813000.0 | |

| | excess_mortality_cumulative_absolute | excess_mortality_cumulative | \ |
|---|---|---|---|
| 116679 | 122180.83 | 6.64 | |
| 116680 | NaN | NaN | |
| 116681 | NaN | NaN | |
| 116682 | NaN | NaN | |
| 116683 | NaN | NaN | |

```
        excess_mortality  excess_mortality_cumulative_per_million
116679             25.88                                 1886.7708
116680               NaN                                       NaN
116681               NaN                                       NaN
116682               NaN                                       NaN
116683               NaN                                       NaN

[5 rows x 67 columns]
```

[181]:
```
igfs_totalcases=igfs_data.groupby(["location"])["hosp_patients"].sum()
```

[180]:
```
igfs_totalcases
```

[180]:
```
location
France      90
Germany      0
Italy      365
Spain      162
Name: hosp_patients, dtype: int64
```

[190]:
```
for column in igfs_data.columns:
    nan_count_igfs=igfs_data[column].isna().sum()
    nan_percentage_igf=round((nan_count_igfs/igfs_data.shape[0])*100,2)
    print(f"{column}contains{nan_count_igfs} NaN values, {nan_percentage_igf}%␣
 ↪of all raws.")
```

```
iso_codecontains0 NaN values, 0.0% of all raws.
continentcontains0 NaN values, 0.0% of all raws.
locationcontains0 NaN values, 0.0% of all raws.
datecontains0 NaN values, 0.0% of all raws.
total_casescontains0 NaN values, 0.0% of all raws.
new_casescontains533 NaN values, 36.51% of all raws.
new_cases_smoothedcontains533 NaN values, 36.51% of all raws.
total_deathscontains0 NaN values, 0.0% of all raws.
new_deathscontains533 NaN values, 36.51% of all raws.
new_deaths_smoothedcontains533 NaN values, 36.51% of all raws.
total_cases_per_millioncontains0 NaN values, 0.0% of all raws.
new_cases_per_millioncontains533 NaN values, 36.51% of all raws.
new_cases_smoothed_per_millioncontains533 NaN values, 36.51% of all raws.
total_deaths_per_millioncontains0 NaN values, 0.0% of all raws.
new_deaths_per_millioncontains533 NaN values, 36.51% of all raws.
new_deaths_smoothed_per_millioncontains533 NaN values, 36.51% of all raws.
reproduction_ratecontains1452 NaN values, 99.45% of all raws.
icu_patientscontains666 NaN values, 45.62% of all raws.
icu_patients_per_millioncontains666 NaN values, 45.62% of all raws.
hosp_patientscontains843 NaN values, 57.74% of all raws.
```

hosp_patients_per_millioncontains843 NaN values, 57.74% of all raws.
weekly_icu_admissionscontains667 NaN values, 45.68% of all raws.
weekly_icu_admissions_per_millioncontains667 NaN values, 45.68% of all raws.
weekly_hosp_admissionscontains665 NaN values, 45.55% of all raws.
weekly_hosp_admissions_per_millioncontains665 NaN values, 45.55% of all raws.
total_testscontains1460 NaN values, 100.0% of all raws.
new_testscontains1460 NaN values, 100.0% of all raws.
total_tests_per_thousandcontains1460 NaN values, 100.0% of all raws.
new_tests_per_thousandcontains1460 NaN values, 100.0% of all raws.
new_tests_smoothedcontains1460 NaN values, 100.0% of all raws.
new_tests_smoothed_per_thousandcontains1460 NaN values, 100.0% of all raws.
positive_ratecontains1460 NaN values, 100.0% of all raws.
tests_per_casecontains1460 NaN values, 100.0% of all raws.
tests_unitscontains1460 NaN values, 100.0% of all raws.
total_vaccinationscontains858 NaN values, 58.77% of all raws.
people_vaccinatedcontains858 NaN values, 58.77% of all raws.
people_fully_vaccinatedcontains858 NaN values, 58.77% of all raws.
total_boosterscontains858 NaN values, 58.77% of all raws.
new_vaccinationscontains885 NaN values, 60.62% of all raws.
new_vaccinations_smoothedcontains708 NaN values, 48.49% of all raws.
total_vaccinations_per_hundredcontains858 NaN values, 58.77% of all raws.
people_vaccinated_per_hundredcontains858 NaN values, 58.77% of all raws.
people_fully_vaccinated_per_hundredcontains858 NaN values, 58.77% of all raws.
total_boosters_per_hundredcontains858 NaN values, 58.77% of all raws.
new_vaccinations_smoothed_per_millioncontains708 NaN values, 48.49% of all raws.
new_people_vaccinated_smoothedcontains708 NaN values, 48.49% of all raws.
new_people_vaccinated_smoothed_per_hundredcontains708 NaN values, 48.49% of all raws.
stringency_indexcontains1460 NaN values, 100.0% of all raws.
population_densitycontains0 NaN values, 0.0% of all raws.
median_agecontains0 NaN values, 0.0% of all raws.
aged_65_oldercontains0 NaN values, 0.0% of all raws.
aged_70_oldercontains0 NaN values, 0.0% of all raws.
gdp_per_capitacontains0 NaN values, 0.0% of all raws.
extreme_povertycontains730 NaN values, 50.0% of all raws.
cardiovasc_death_ratecontains0 NaN values, 0.0% of all raws.
diabetes_prevalencecontains0 NaN values, 0.0% of all raws.
female_smokerscontains0 NaN values, 0.0% of all raws.
male_smokerscontains0 NaN values, 0.0% of all raws.
handwashing_facilitiescontains1460 NaN values, 100.0% of all raws.
hospital_beds_per_thousandcontains0 NaN values, 0.0% of all raws.
life_expectancycontains0 NaN values, 0.0% of all raws.
human_development_indexcontains0 NaN values, 0.0% of all raws.
populationcontains0 NaN values, 0.0% of all raws.
excess_mortality_cumulative_absolutecontains1248 NaN values, 85.48% of all raws.
excess_mortality_cumulativecontains1248 NaN values, 85.48% of all raws.
excess_mortalitycontains1248 NaN values, 85.48% of all raws.
excess_mortality_cumulative_per_millioncontains1248 NaN values, 85.48% of all

raws.

**3.9 La colonna hosp_patients presenta 843 valori nulli: "hosp_patientscontains 843 NaN values, 57.74% of all raws". In questo caso di può procedere alla eliminazione e/o ssotituzione dei dati nulli al fine di poter effettuare una analisi accurata.**

[ ]: