```
# Importing libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import urllib.request, json
```

In [2]:
```
# Several methods to retrieve JSON
# df=pd.read_json('https://www..../.json')
# data = json.load('/home/.../.json')

# with urllib.request.urlopen("https://.../.json") as url:
#     data = json.load(url)
#     print(data)

# with open('/.../Covid19.json', 'r') as myfile:
#     data=myfile.read()

# # parse file
# obj = json.loads(data)
```

In [3]:
```
df=pd.read_json('/.../Covid19.json')
```

In [4]:
```
### The dataset below is as of 10-02-2023.
```

In [5]:
```
# Showing information form JSON file, Here is the dataset we have to work on
df
```

Out[5]:

	sno	state_name	active	positive	cured	death	new_active	new_positive	new_cu
0	1	dolnośląskie	0	10751	10622	129	0	10751	106
1	2	kujawsko-pomorskie	4	2339098	2324361	14733	3	2339098	23243
2	3	lubelskie	0	66891	66595	296	0	66891	665
3	4	łódzkie	0	746100	738065	8035	0	746100	7380
4	5	małopolskie	3	851428	839122	12303	3	851428	8391
5	6	mazowieckie	4	99373	98187	1182	6	99375	981
6	7	opolskie	6	1177805	1163653	14146	6	1177806	11636
7	8	podkarpackie	0	11591	11587	4	0	11591	177
8	9	podlaskie	38	2007692	1981131	26523	62	2007718	19811
9	10	pomorskie	32	259191	255146	4013	40	259199	2551
10	11	śląskie	233	1278016	1266736	11047	268	1278074	12667
11	12	świętokrzyskie	49	1056797	1046034	10714	47	1056803	10460
12	13	warmińsko-mazurskie	60	312858	308584	4214	100	312900	3085
13	14	wielkopolskie	41	479527	474701	4785	43	479533	4747
14	15	zachodniopomorskie	2	442579	437246	5331	4	442581	4372

In [6]:
```
# In this file are 15 rows and 14 columns
df.shape
```

Out[6]:
```
(15, 14)
```

In [7]:
```
# In this file type of data present in the columns(int,object)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   sno                15 non-null     int64 
 1   state_name         15 non-null     object
 2   active             15 non-null     int64 
 3   positive           15 non-null     int64 
 4   cured              15 non-null     int64 
 5   death              15 non-null     int64 
 6   new_active         15 non-null     int64 
 7   new_positive       15 non-null     int64 
 8   new_cured          15 non-null     int64 
 9   new_death          15 non-null     int64 
 10  death_reconsille   15 non-null     object
 11  total              15 non-null     object
 12  state_code         15 non-null     int64 
 13  actualdeath24hrs   15 non-null     int64 
dtypes: int64(11), object(3)
memory usage: 1.8+ KB
```

In … *# Viewing the descriptive statistics of the data like mean, std deviation, min and max values present in the*

```
df.describe()
```

Out[8]:

	sno	active	positive	cured	death	new_active	new_positiv
count	15.000000	15.000000	1.500000e+01	1.500000e+01	15.000000	15.000000	1.500000e+0
mean	8.000000	31.466667	7.426465e+05	7.347847e+05	7830.333333	38.800000	7.426565e+0
std	4.472136	59.562532	7.198113e+05	7.132340e+05	7272.052763	70.127639	7.198153e+0
min	1.000000	0.000000	1.075100e+04	1.062200e+04	4.000000	0.000000	1.075100e+0
25%	4.500000	1.000000	1.792820e+05	1.766665e+05	2597.500000	1.500000	1.792870e+0
50%	8.000000	4.000000	4.795270e+05	4.747010e+05	5331.000000	6.000000	4.795330e+0
75%	11.500000	39.500000	1.117301e+06	1.104844e+06	11675.000000	45.000000	1.117304e+0
max	15.000000	233.000000	2.339098e+06	2.324361e+06	26523.000000	268.000000	2.339098e+0

In [9]: *# Displaying all the column names present in data*

```
df.columns
```

Out[9]:

```
Index(['sno', 'state_name', 'active', 'positive', 'cured', 'death',
       'new_active', 'new_positive', 'new_cured', 'new_death',
       'death_reconsille', 'total', 'state_code', 'actualdeath24hrs'],
      dtype='object')
```

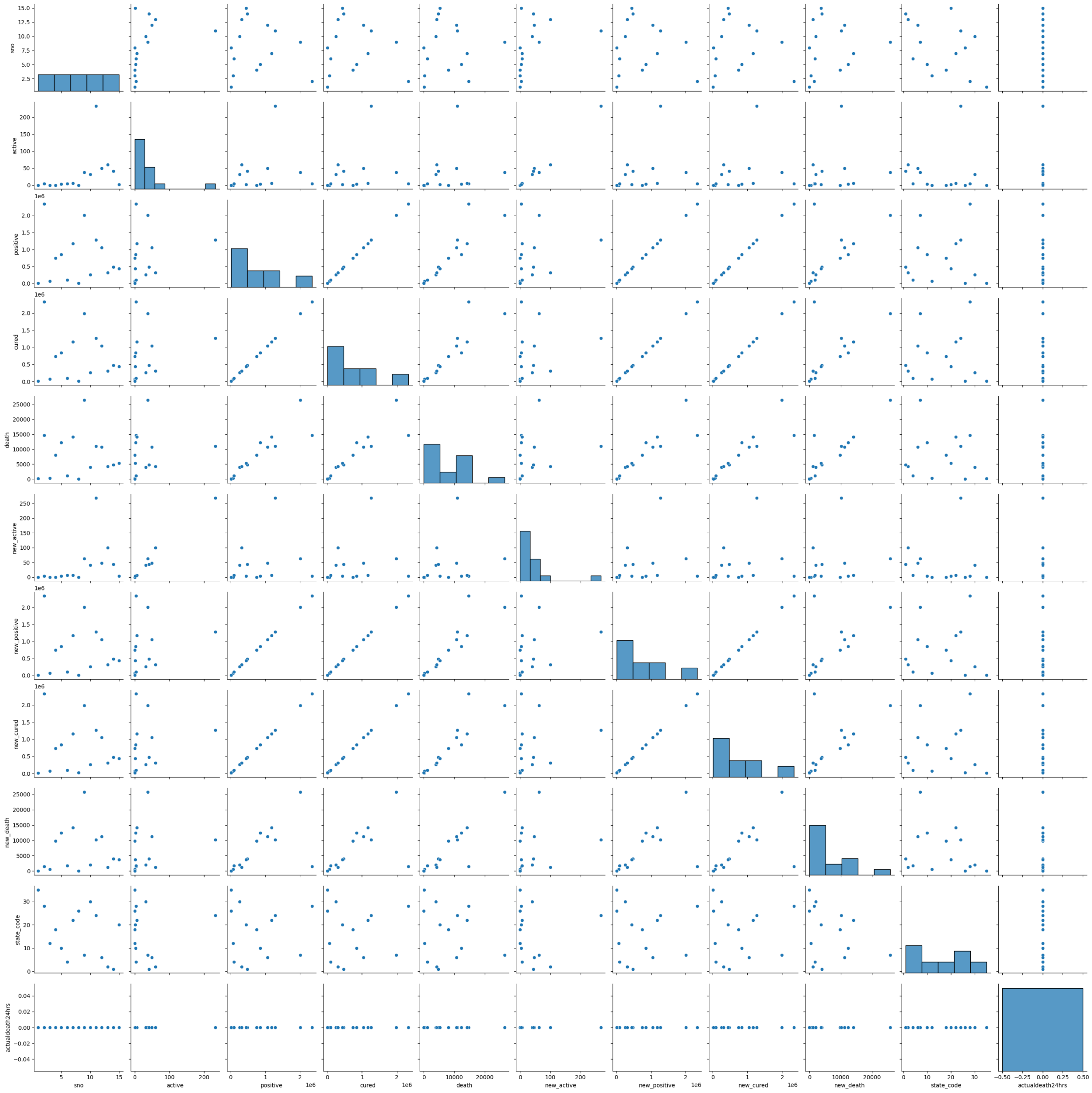
In [10]: *# Display the last 6 rows*

```
df.tail(6)
```

Out[10]:

	sno	state_name	active	positive	cured	death	new_active	new_positive	new_cu
9	10	pomorskie	32	259191	255146	4013	40	259199	2551
10	11	śląskie	233	1278016	1266736	11047	268	1278074	12667
11	12	świętokrzyskie	49	1056797	1046034	10714	47	1056803	1046(
12	13	warmińsko-mazurskie	60	312858	308584	4214	100	312900	3085
13	14	wielkopolskie	41	479527	474701	4785	43	479533	4747
14	15	zachodniopomorskie	2	442579	437246	5331	4	442581	4372

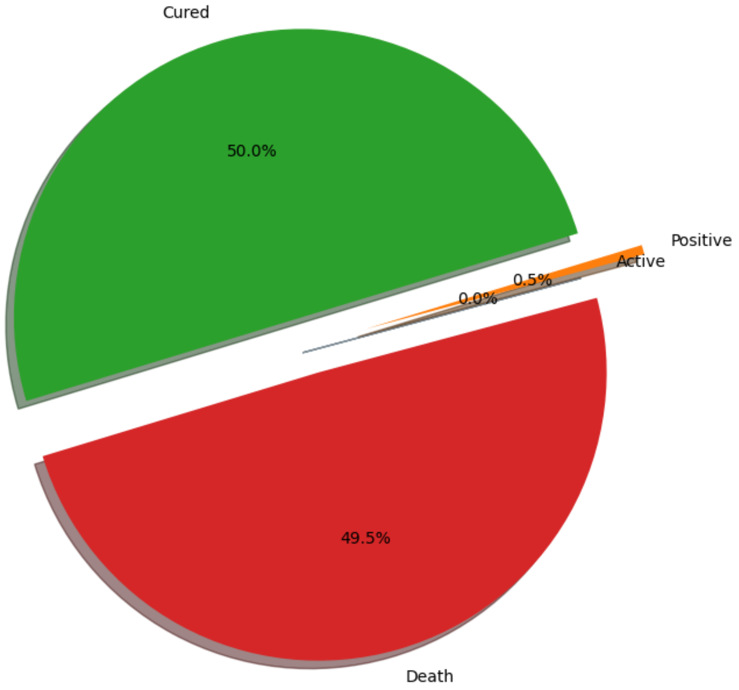
```
In [11]:# The total number of active cases in Poland 472
         df['active'].sum(axis = 0)

   Out[11]:472
In [12]:# The total number of deaths in Poland 117.455
         df['death'].sum(axis = 0)

   Out[12]:117455
In [13]:# The total number of positive in Poland 11.139.697
         df['positive'].sum(axis = 0)

   Out[13]:11139697
In [14]:# The total number of cured in Poland 11.021.770
         df['cured'].sum(axis = 0)

   Out[14]:11021770
In [15]:# The total number of new_active cases in Poland 582
         df['new_active'].sum(axis = 0)

   Out[15]:582
In [16]:# The total number of new_deaths in Poland 98.483
         df['new_death'].sum(axis = 0)

   Out[16]:98483
In [17]:# The total number of new_positive in Poland 11.139.848
         df['new_positive'].sum(axis = 0)

   Out[17]:11139848
In [18]:# The total number of new_cured in Poland 11.027.971
         df['new_cured'].sum(axis = 0)

   Out[18]:11027971
In [19]:### Data Visualization
In [20]:# Plotting scatter plots of all data
         sns.pairplot(data=df)
         # plt.show()

   Out[20]:<seaborn.axisgrid.PairGrid at 0x7fc8ad3e5900>
```



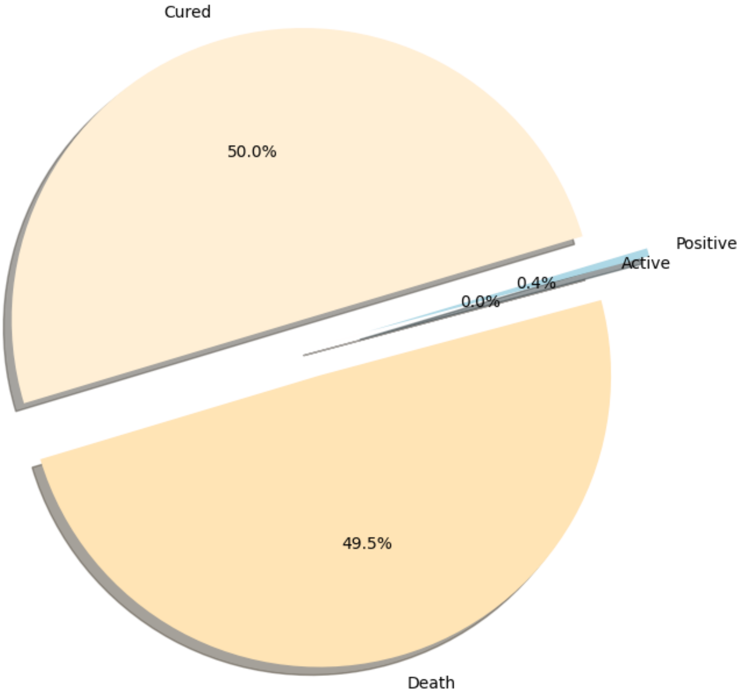
```
In [21]:# Storing total cases
        total_df = df.sum()
        total_df
```

```
Out[21]:sno                                                          120
        state_name          dolnośląskiekujawsko-pomorskielubelskiełódzkie...
        active                                                       472
        positive                                                11139697
        cured                                                   11021770
        death                                                     117455
        new_active                                                   582
        new_positive                                            11139848
        new_cured                                               11027971
        new_death                                                  98483
        death_reconsille
        total
        state_code                                                   245
        actualdeath24hrs                                               0
        dtype: object
```

```
In [… # Chart visualization
      my_data = [472,117455,11139697,11021770]
      my_labels = 'Active','Positive','Cured','Death'
      my_explode = (0,0.2,0.1,0.1)
      fig1, ax1 = plt.subplots(figsize=(13, 8))
      plt.pie(my_data, labels=my_labels, autopct='%1.1f%%', startangle=15, shadow = True, explode=my_ex
      plt.axis('equal')
      plt.show()
```

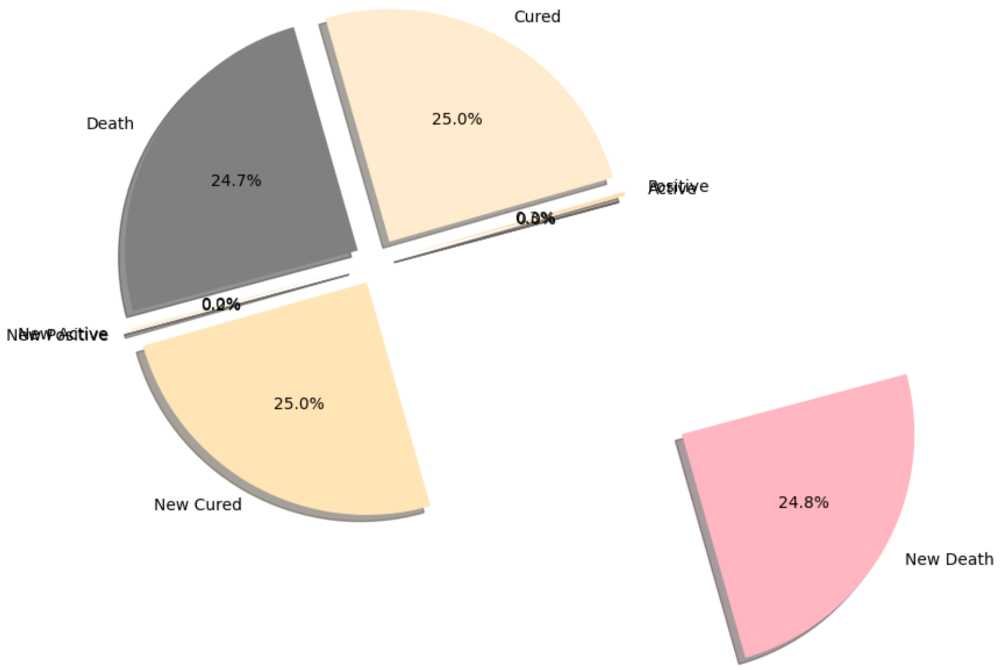


```
In […# Chart visualization
    my_data = [582,98483,11139848,11027971]
    my_labels = 'Active','Positive','Cured','Death'
    my_explode = (0,0.2,0.1,0.1)
    my_colors = ['blanchedalmond','lightblue','papayawhip','moccasin']
    fig1, ax1 = plt.subplots(figsize=(13, 8))
    plt.pie(my_data, labels=my_labels, autopct='%1.1f%%', startangle=15, shadow = True, colors=my_colors
    plt.axis('equal')
    plt.show()
```

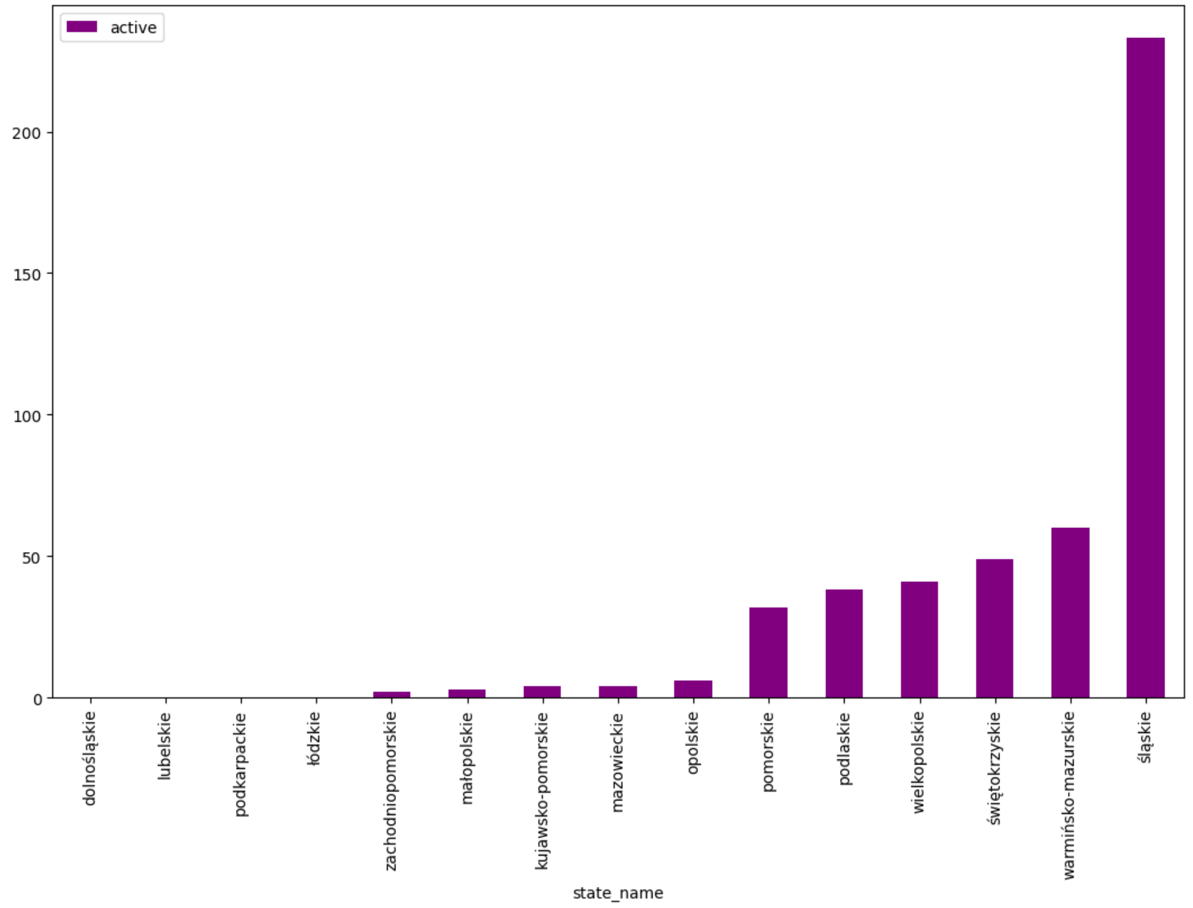


```
In […# Chart visualization
    my_data = [472,117455,11139697,11021770,582,98483,11139848,11027971]
    my_labels = 'Active','Positive','Cured','Death','New Active','New Positive','New Cured','New Death'
    my_explode = (0.1,0.1,0.1,0.1,0.1,0.1,0.1,1.5)
    my_colors = ['gray','navajowhite','blanchedalmond','grey','lightblue','papayawhip','moccasin','lightpink']
    fig1, ax1 = plt.subplots(figsize=(13, 8))
```

```
        plt.pie(my_data, labels=my_labels, autopct='%1.1f%%', startangle=15, shadow = True, colors=my_color
        plt.axis('equal')
        plt.show()
```

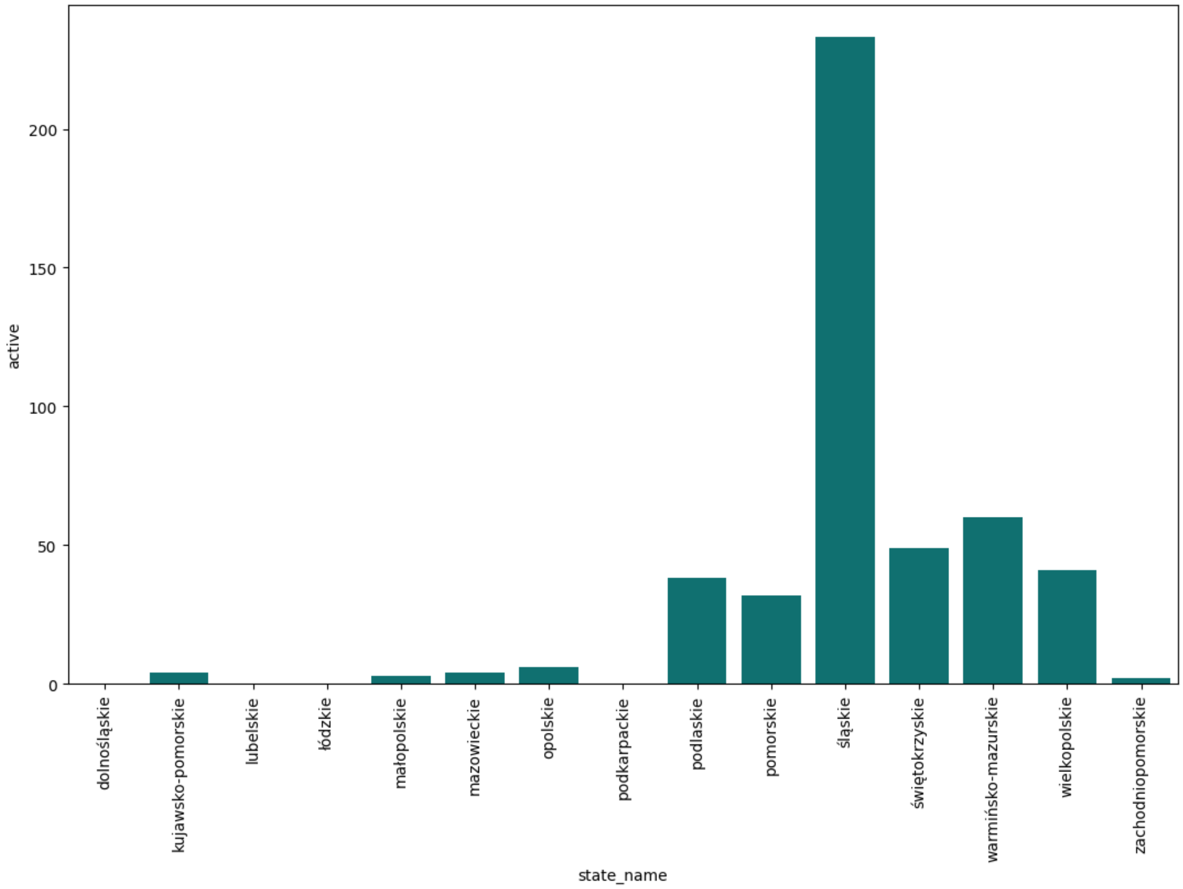


```
In [25]:# Figure size
        plt.rcParams['figure.figsize']=(13,8)
In [2…# We get a picture of the states in an increasing order based on their active level of cases.
        df[['state_name','active']].groupby(["state_name"]).mean().sort_values(by='active').plot.bar(color='purple
        plt.show()
```



```
In [27]:#  Number of active cases
        plt.figure(figsize=(13, 8))
        plt.xticks(rotation=90)
```

```
sns.barplot(x='state_name',y='active',color='teal',data=df);
plt.show()
```



```
In [28]:df.columns
```

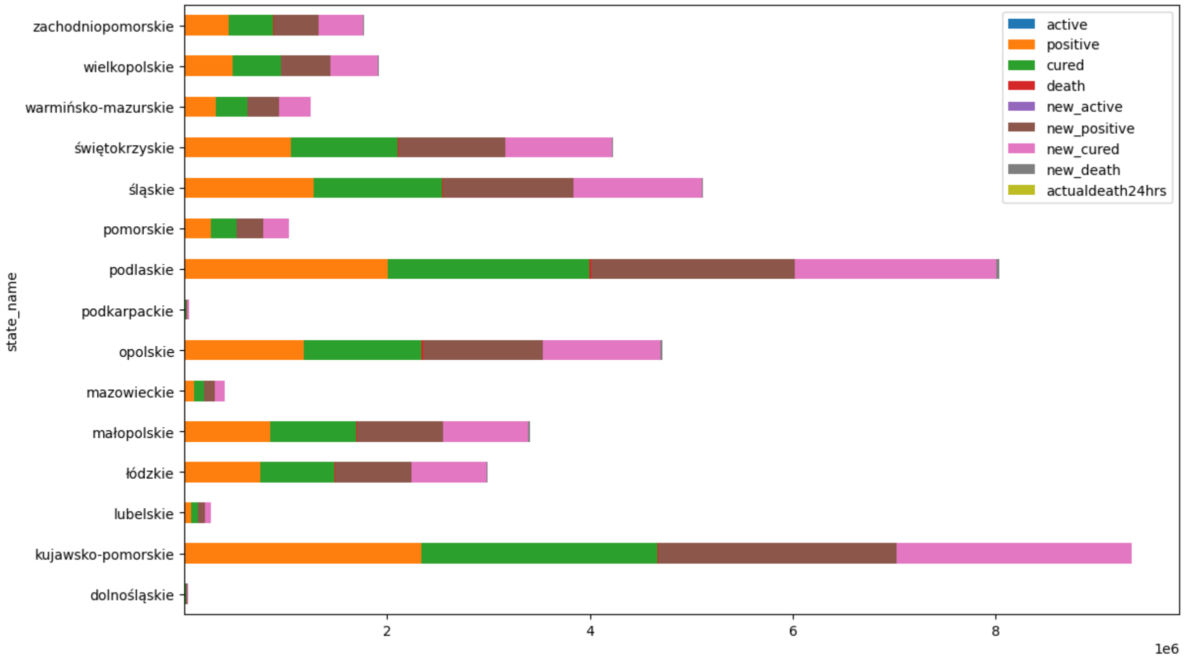
```
Out[28]:Index(['sno', 'state_name', 'active', 'positive', 'cured', 'death',
                'new_active', 'new_positive', 'new_cured', 'new_death',
                'death_reconsille', 'total', 'state_code', 'actualdeath24hrs'],
               dtype='object')
```

```
In [29]:df.drop(['sno','state_code'],axis=1,inplace=True)
In [30]:df.head(5)
```

Out[30]:

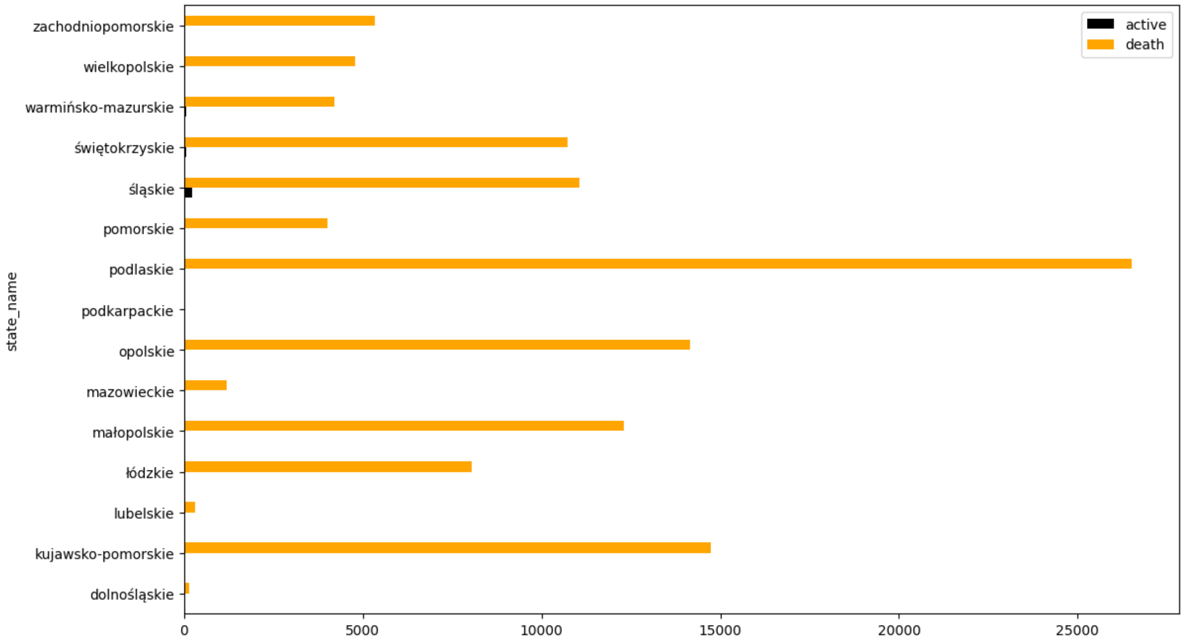
	state_name	active	positive	cured	death	new_active	new_positive	new_cured	new_deat
0	dolnośląskie	0	10751	10622	129	0	10751	10622	1
1	kujawsko-pomorskie	4	2339098	2324361	14733	3	2339098	2324362	147
2	lubelskie	0	66891	66595	296	0	66891	66595	52
3	łódzkie	0	746100	738065	8035	0	746100	738065	985
4	małopolskie	3	851428	839122	12303	3	851428	839122	1244

```
In [31]:df=df.set_index('state_name')
In [37]:# Stacked bar plot
        df.plot.barh(stacked=True,figsize=(13,8))
        plt.show()
```



In [40]:

```
# Bar plot for active and death cases
df1=df[['active', 'death']]
df1.plot.barh(color={"active": "black", "death": "orange"},figsize=(13,8))
plt.show()
```


state_name
zachodniopomorskie
wielkopolskie
warmińsko-mazurskie
świętokrzyskie
śląskie
pomorskie
podlaskie
podkarpackie
opolskie
mazowieckie
małopolskie
łódzkie
lubelskie
kujawsko-pomorskie
dolnośląskie
active
death