Wczytanie danych

In [1]:
```
import pandas as pd
df = pd.read_csv("IHME_HEALTH_SPENDING_1995_2018_Y2021M09D22.CSV", encoding='latin-1', e

df.head()
```

Out[1]:

| | location_id | location_name | iso3 | level | year | the_total_mean | the_total_lower | the_total_upper | the_total_ppp |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 160 | Afghanistan | AFG | Country | 1995 | 528409 | 417121 | 665425 | 2 |
| **1** | 160 | Afghanistan | AFG | Country | 1996 | 516915 | 412035 | 651034 | 2 |
| **2** | 160 | Afghanistan | AFG | Country | 1997 | 509874 | 413424 | 636436 | 2 |
| **3** | 160 | Afghanistan | AFG | Country | 1998 | 485561 | 394629 | 605041 | 2 |
| **4** | 160 | Afghanistan | AFG | Country | 1999 | 463720 | 382723 | 564170 | 1 |

5 rows × 84 columns

Załadowanie matplotlib

In [2]:
```
import matplotlib.pyplot as plt
```

Podgląd danych

In [3]:
```
df["location_name"].unique()
```

Out[3]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra',
       'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bermuda', 'Bhutan', 'Bolivia (Plurinational State of)',
       'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'Brunei Darussalam', 'Bulgaria', 'Burkina Faso', 'Burundi',
       'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada',
       'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia',
       'Comoros', 'Congo', 'Cook Islands', 'Costa Rica', 'Croatia',
       'Cuba', 'Cyprus', 'Czechia', "Côte d'Ivoire",
       "Democratic People's Republic of Korea",
       'Democratic Republic of the Congo', 'Denmark', 'Djibouti',
       'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
       'Eswatini', 'Ethiopia', 'Fiji', 'Finland', 'France', 'Gabon',
       'Gambia', 'Georgia', 'Germany', 'Ghana', 'Greece', 'Greenland',
       'Grenada', 'Guam', 'Guatemala', 'Guinea', 'Guinea-Bissau',
       'Guyana', 'Haiti', 'Honduras', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran (Islamic Republic of)', 'Iraq', 'Ireland',
       'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan',
       'Kenya', 'Kiribati', 'Kuwait', 'Kyrgyzstan',
       "Lao People's Democratic Republic", 'Latvia', 'Lebanon', 'Lesotho',
       'Liberia', 'Libya', 'Lithuania', 'Luxembourg', 'Madagascar',
       'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta',
       'Marshall Islands', 'Mauritania', 'Mauritius', 'Mexico',
       'Micronesia (Federated States of)', 'Monaco', 'Mongolia',
       'Montenegro', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
       'Nauru', 'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua',
       'Niger', 'Nigeria', 'Niue', 'North Macedonia',
       'Northern Mariana Islands', 'Norway', 'Oman', 'Pakistan', 'Palau',
       'Palestine', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru',
       'Philippines', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',
```

```
'Republic of Korea', 'Republic of Moldova', 'Romania',
'Russian Federation', 'Rwanda', 'Saint Kitts and Nevis',
'Saint Lucia', 'Saint Vincent and the Grenadines', 'Samoa',
'San Marino', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal',
'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia',
'Slovenia', 'Solomon Islands', 'Somalia', 'South Africa',
'South Sudan', 'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Sweden',
'Switzerland', 'Syrian Arab Republic',
'Taiwan (Province of China)', 'Tajikistan', 'Thailand',
'Timor-Leste', 'Togo', 'Tokelau', 'Tonga', 'Trinidad and Tobago',
'Tunisia', 'Turkey', 'Turkmenistan', 'Tuvalu', 'Uganda', 'Ukraine',
'United Arab Emirates', 'United Kingdom',
'United Republic of Tanzania', 'United States Virgin Islands',
'United States of America', 'Uruguay', 'Uzbekistan', 'Vanuatu',
'Venezuela (Bolivarian Republic of)', 'Viet Nam', 'Yemen',
'Zambia', 'Zimbabwe', 'Global', 'High income', 'Low income',
'Lower-middle income', 'Upper-middle income',
'Central Europe, Eastern Europe, and Central Asia', 'High-income',
'Latin America and Caribbean', 'North Africa and Middle East',
'South Asia', 'Southeast Asia, East Asia, and Oceania',
'Sub-Saharan Africa'], dtype=object)
```

In [149…

```
d1 = df[df["location_name"] == "Sierra Leone"]
d1[["year","the_total_mean"]]
```

Out[149]:

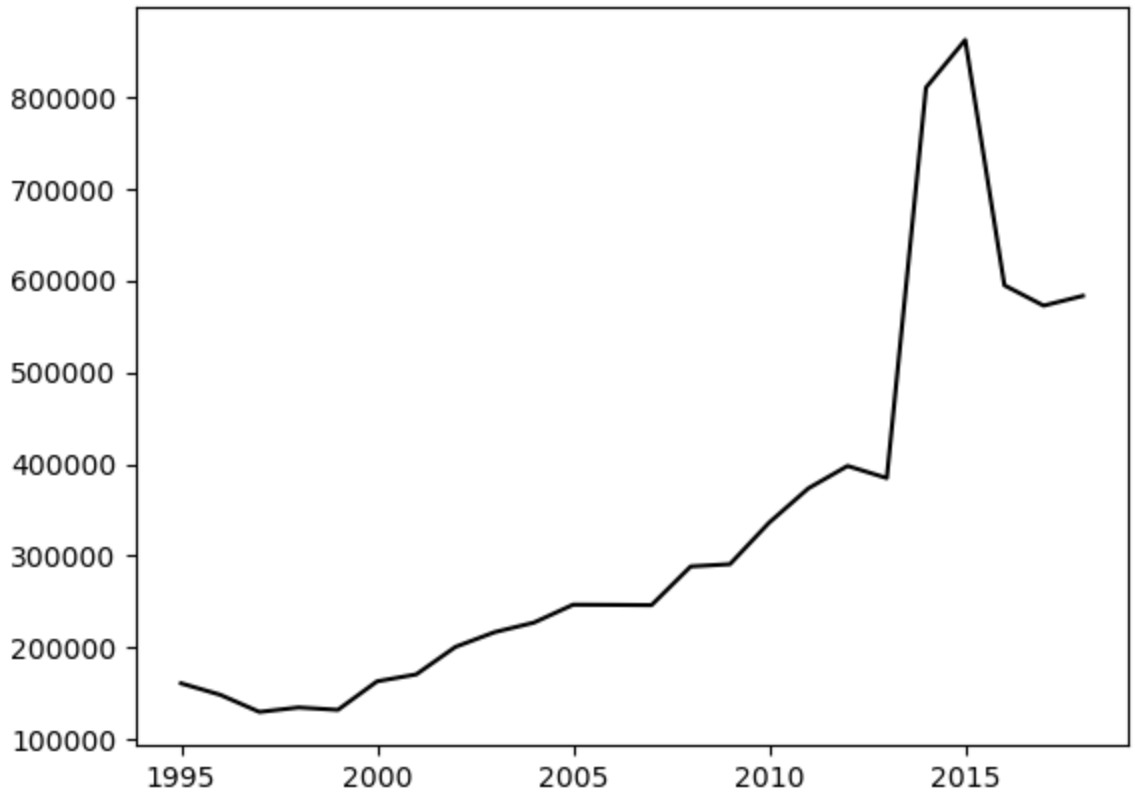
| | year | the_total_mean |
|---|---|---|
| **3888** | 1995 | 161382 |
| **3889** | 1996 | 148878 |
| **3890** | 1997 | 130216 |
| **3891** | 1998 | 135092 |
| **3892** | 1999 | 132452 |
| **3893** | 2000 | 163519 |
| **3894** | 2001 | 171104 |
| **3895** | 2002 | 201043 |
| **3896** | 2003 | 217092 |
| **3897** | 2004 | 227436 |
| **3898** | 2005 | 246956 |
| **3899** | 2006 | 246768 |
| **3900** | 2007 | 246467 |
| **3901** | 2008 | 288623 |
| **3902** | 2009 | 291038 |
| **3903** | 2010 | 336302 |
| **3904** | 2011 | 373600 |
| **3905** | 2012 | 397972 |
| **3906** | 2013 | 384762 |
| **3907** | 2014 | 809772 |
| **3908** | 2015 | 861612 |
| **3909** | 2016 | 594553 |

| | | |
|---|---|---|
| **3910** | 2017 | 572442 |
| **3911** | 2018 | 583077 |

Wykres - średnia wydawanych pieniędzy na służbę zdrowia w danym roku w Sierra Leone (x, y, styl linii)

In [150…
```
plt.plot(h["year"], d1["the_total_mean"], 'k-')
```

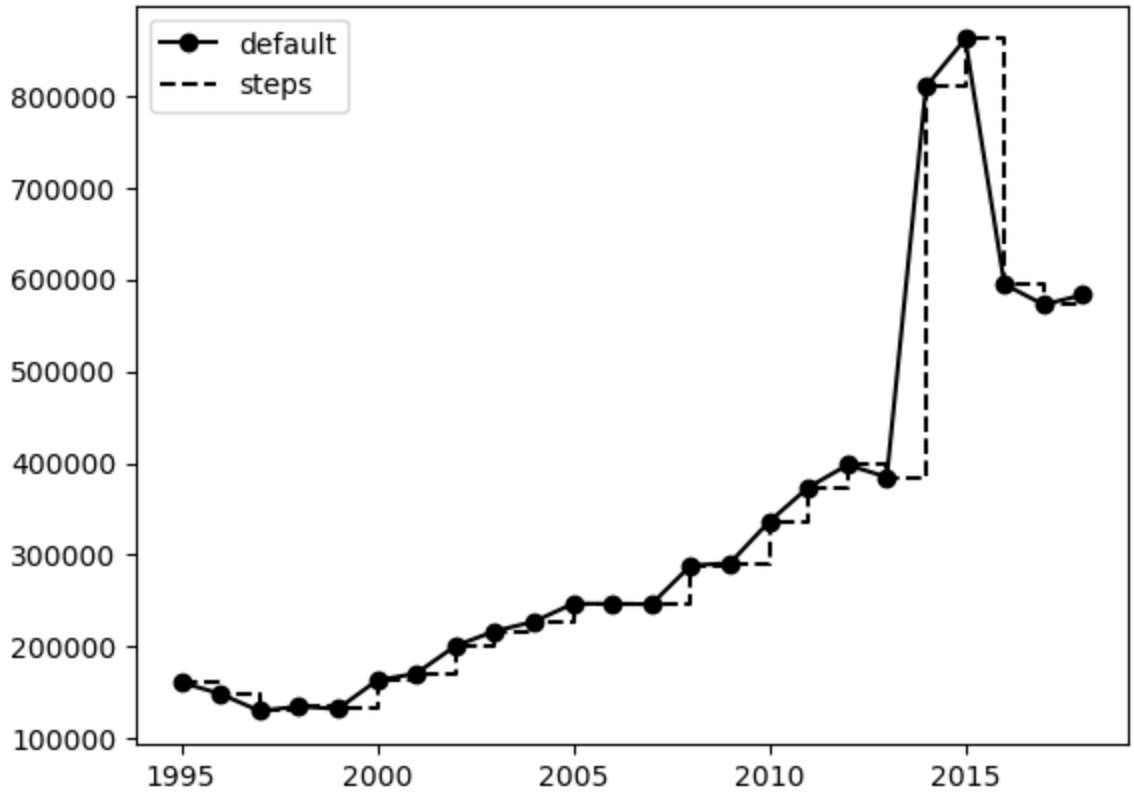
Out[150]:
```
[<matplotlib.lines.Line2D at 0x16495431790>]
```




Różne style rysowania

In [151…
```
plt.plot(d1["year"], d1["the_total_mean"], 'ko-', label="default")
plt.plot(d1["year"], d1["the_total_mean"], 'k--', drawstyle='steps-post', label="steps")
plt.legend()
```

Out[151]:
```
<matplotlib.legend.Legend at 0x16495973e80>
```

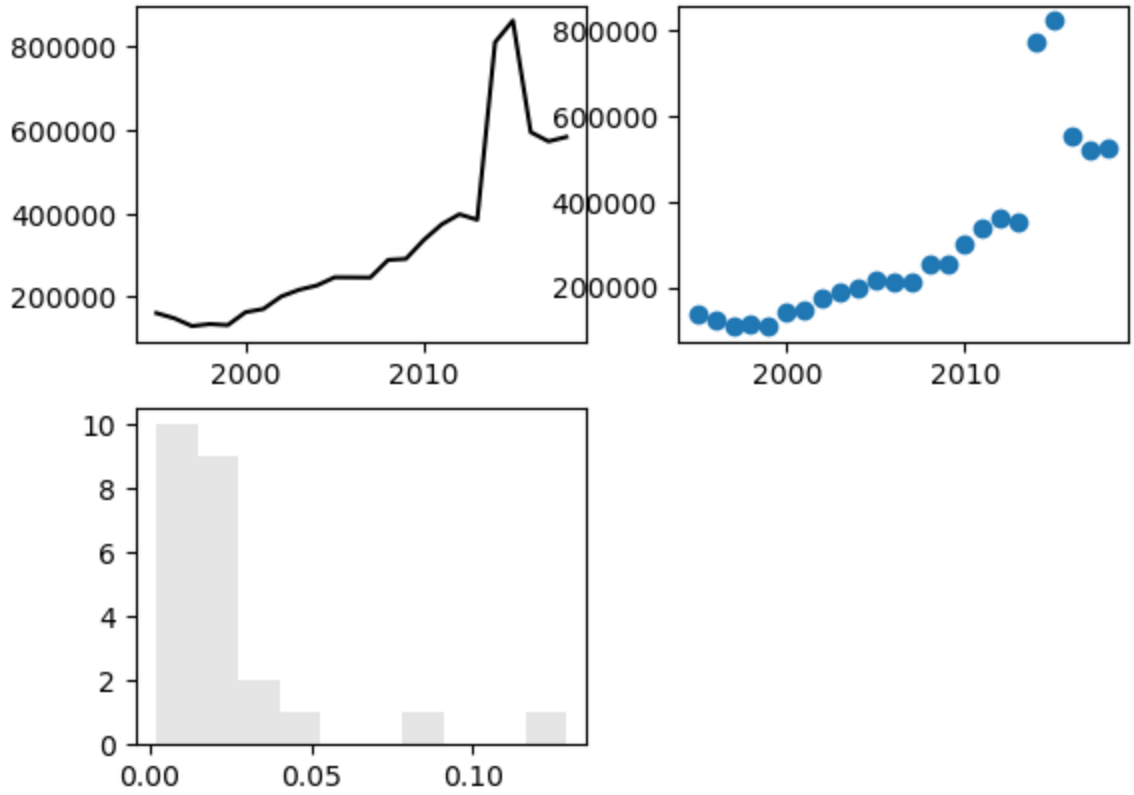



Grupy wykresów

(rozmiar x, rozmiar y, identyfikator)

In [153…
```
fig = plt.figure()

ax1 = fig.add_subplot(2,2,1)
ax1.plot(d1["year"], d1["the_total_mean"], 'k-')

ax2 = fig.add_subplot(2,2,2)
ax2.scatter(d1["year"], d1["the_total_lower"])

ax3 = fig.add_subplot(2,2,3)
ax3.hist(d1["dah_per_gdp_lower"], bins = 10, color = 'k', alpha=0.1)
#bins = ilość "pól" na wykresie
```

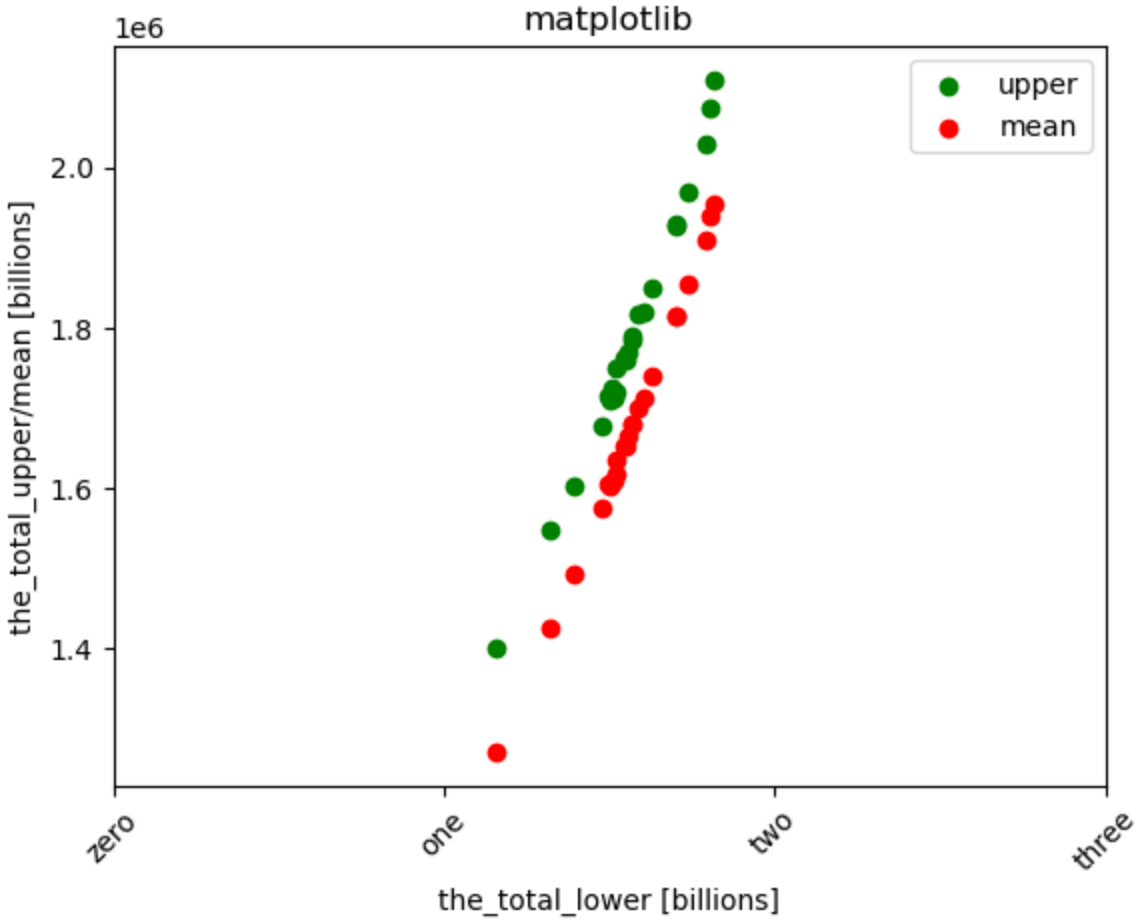
Out[153]:
```
(array([10.,  9.,  2.,  1.,  0.,  0.,  1.,  0.,  0.,  1.]),
 array([0.002 , 0.0147, 0.0274, 0.0401, 0.0528, 0.0655, 0.0782, 0.0909,
        0.1036, 0.1163, 0.129 ]),
 <BarContainer object of 10 artists>)
```




Legenda i etykiety osi

In [154…

```
d2 = df[df["location_name"] == "El Salvador"]

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)

ax.scatter(d2["the_total_lower"], d2["the_total_upper"], color='g', label="upper")
ax.scatter(d2["the_total_lower"], d2["the_total_mean"], color='r', label="mean")
ax.set_xticks([0, 1000000, 2000000, 3000000])
ax.set_xticklabels(['zero', 'one', 'two', 'three'], rotation=45, fontsize='medium')
ax.set_title('matplotlib')
ax.set_xlabel('the_total_lower [billions]')
ax.set_ylabel('the_total_upper/mean [billions]')
ax.legend(loc='best')

plt.savefig('figure.png', dpi=400, bbox_inches='tight')
```




Zapis do pliku

(ścieżka, dpi, ucinanie białych fragmentów wokół obrazu)

Zapis musi być wykonany przed wyświetleniem, inaczej obraz będzie pusty - (wyświetlenie czyści wykres)

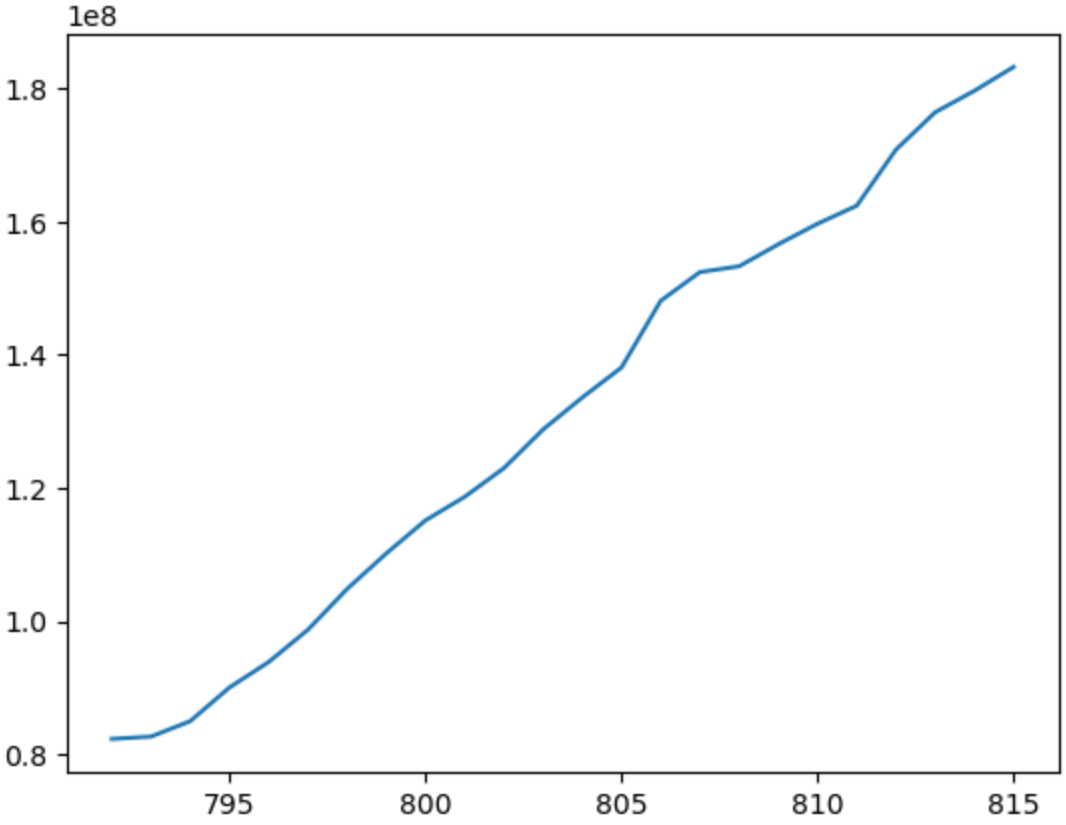
In [9]:
```
plt.savefig('figure.png', dpi=400, bbox_inches='tight')
```

```
<Figure size 640x480 with 0 Axes>
```

Wykresy pandas

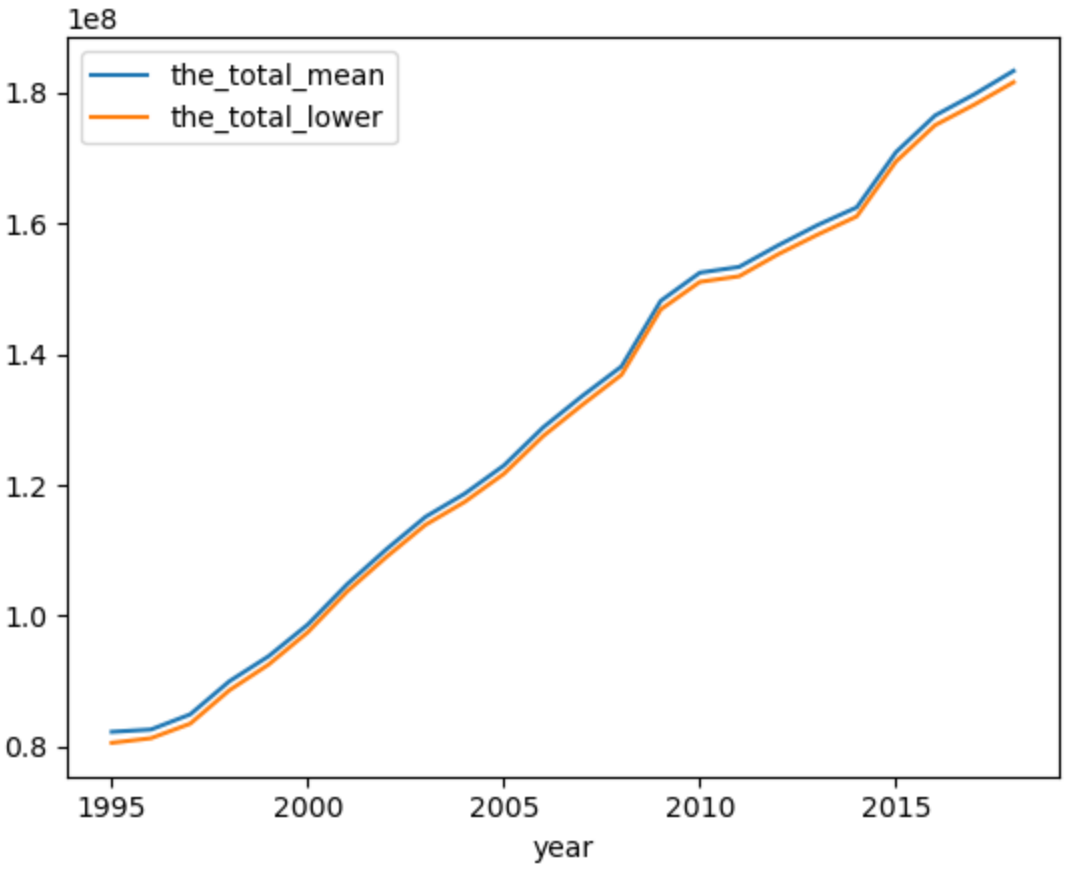
In [155…
```
d3 = df[df["location_name"] == "Canada"]

figure = pd.Series(d3["the_total_mean"])
figure.plot()
```

Out[155]:
```
<AxesSubplot:>
```




In [161…
```
d4 = d3[["the_total_mean", "the_total_lower", "year"]]
figure = d4
figure.plot(x = "year")
```

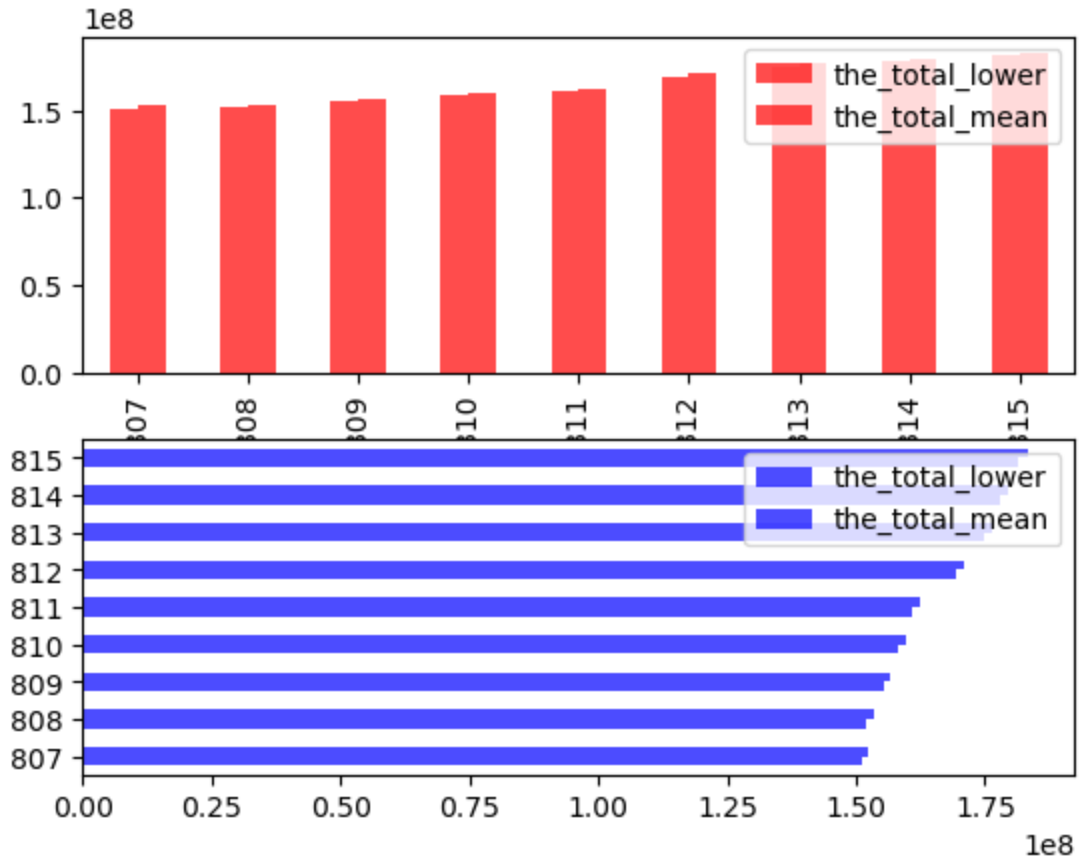
Out[161]:
```
<AxesSubplot:xlabel='year'>
```




Wykresy kolumnowe

In [162…
```
fig, axes = plt.subplots(2, 1)
d5 = d4[d4["year"] > 2009]
d5 = d5[["the_total_lower", "the_total_mean"]]
data = d5
data.plot.bar(ax=axes[0], color='r', alpha=0.7) #rysowanie w pionie
data.plot.barh(ax=axes[1], color='b', alpha=0.7) #rysowanie w poziomie
```

Out[162]:
```
<AxesSubplot:>
```



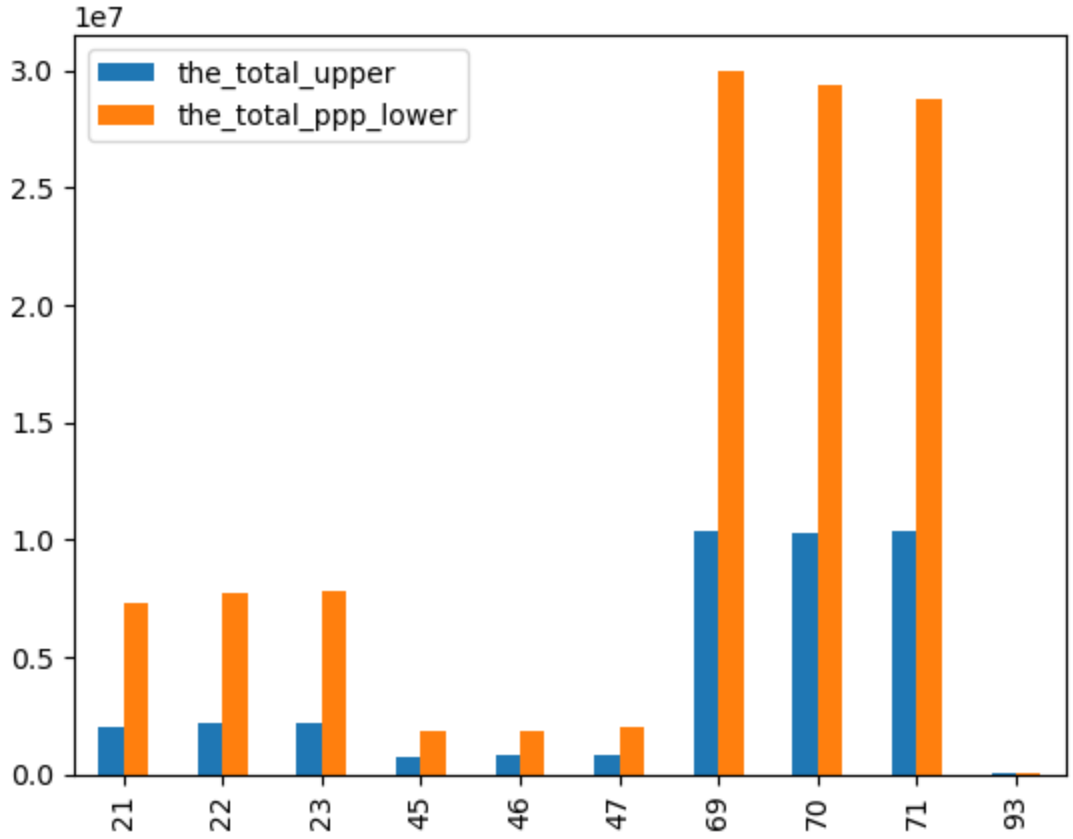

In [165…
```
d6 = df[df["year"] > 2015]
d6 = d6[["the_total_upper", "the_total_ppp_lower"]]
d6 = d6.head(10)
d6
```

Out[165]:

| | the_total_upper | the_total_ppp_lower |
|---|---|---|
| **21** | 2028013 | 7253099 |
| **22** | 2142606 | 7709396 |
| **23** | 2207902 | 7779658 |
| **45** | 738868 | 1795580 |
| **46** | 779525 | 1872224 |
| **47** | 838110 | 1990826 |
| **69** | 10365602 | 30006433 |
| **70** | 10242469 | 29389584 |
| **71** | 10379786 | 28755496 |
| **93** | 36712 | 27535 |

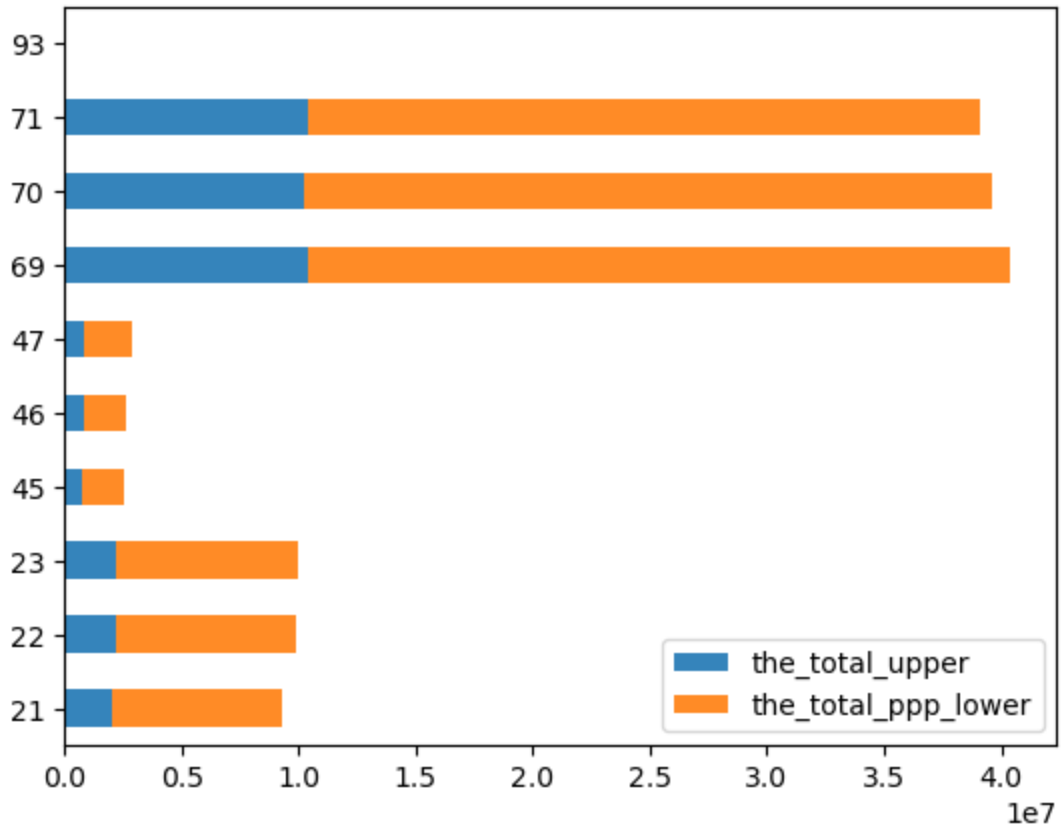
In [166…
```
d6.plot.bar()
```

Out[166]: <AxesSubplot:>




In [167…
```
d6.plot.barh(stacked=True, alpha=0.9)
```

Out[167]: <AxesSubplot:>




Seaborn

In [177…
```
import seaborn as sbn
```

In [178…

```
d7 = df
d7 = d7[d7["year"]> 2006]
d7
```

Out[178]:

| | location_id | location_name | iso3 | level | year | the_total_mean | the_total_lower | the_total_upper | the_total |
|---|---|---|---|---|---|---|---|---|---|
| **12** | 160 | Afghanistan | AFG | Country | 2007 | 1040372 | 956086 | 1141413 | |
| **13** | 160 | Afghanistan | AFG | Country | 2008 | 1066858 | 982415 | 1161963 | |
| **14** | 160 | Afghanistan | AFG | Country | 2009 | 1236730 | 1149054 | 1337448 | |
| **15** | 160 | Afghanistan | AFG | Country | 2010 | 1262196 | 1166456 | 1364153 | |
| **16** | 160 | Afghanistan | AFG | Country | 2011 | 1373092 | 1273961 | 1482720 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **5227** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2014 | 71349683 | 69713253 | 73018691 | |
| **5228** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2015 | 73465621 | 71828527 | 75186691 | |
| **5229** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2016 | 73972108 | 72267558 | 75733371 | |
| **5230** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2017 | 77406545 | 75367505 | 79506397 | |
| **5231** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2018 | 78994327 | 76466329 | 81632889 | |

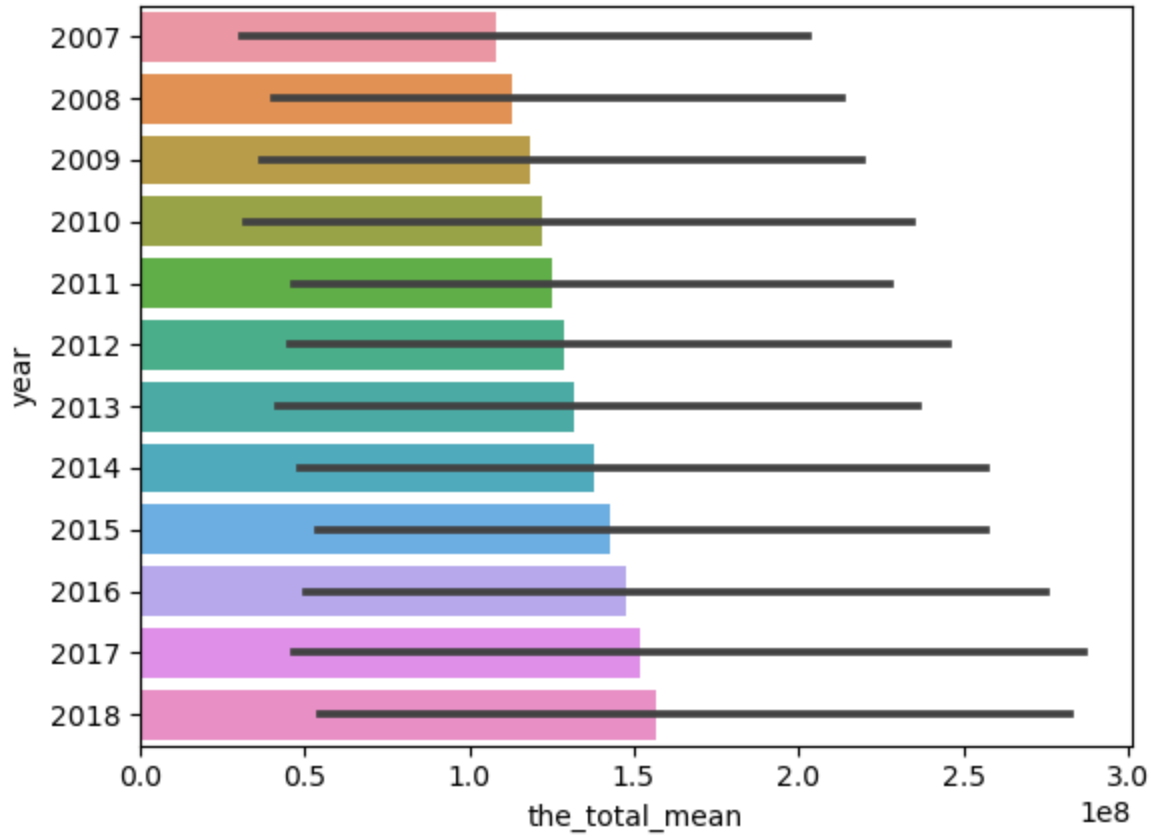
2616 rows × 84 columns

In [179…

```
sbn.barplot(x='the_total_mean', y='year', data=d7, orient='h')
#automatycznie wyświetla przedział ufności
```

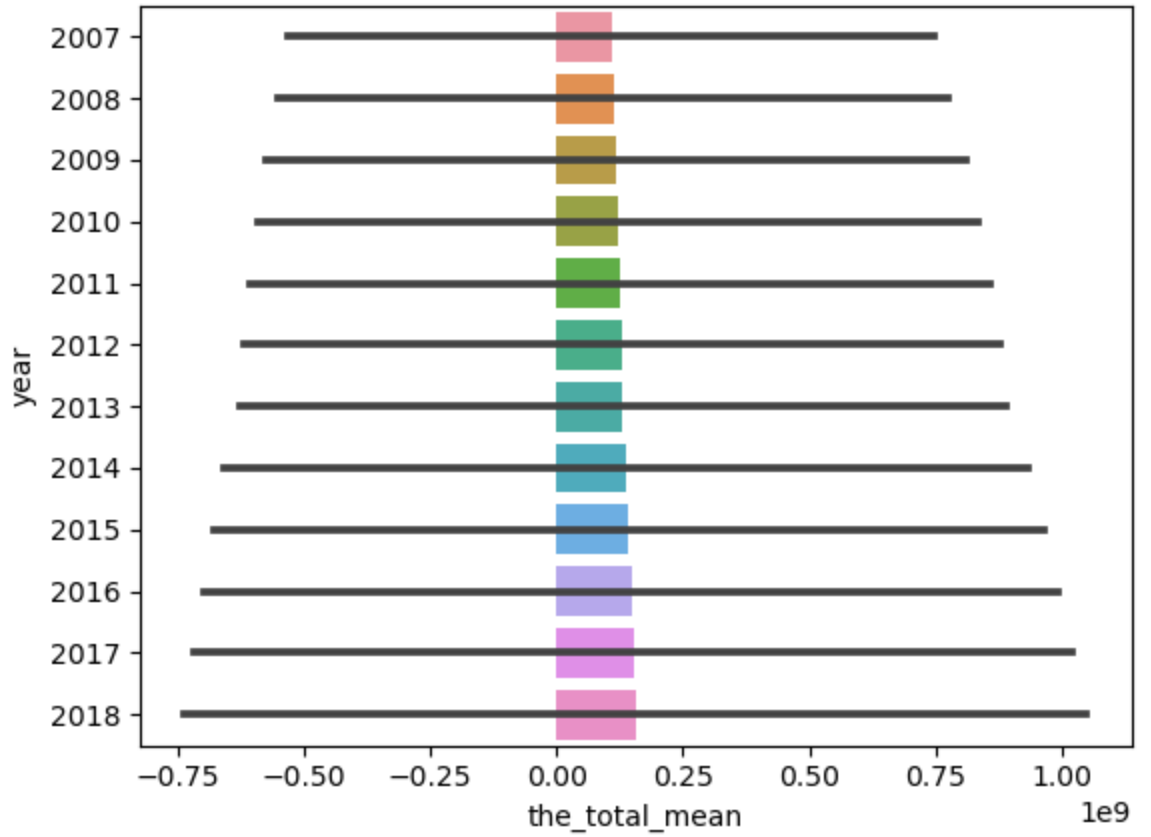
Out[179]:

```
<AxesSubplot:xlabel='the_total_mean', ylabel='year'>
```




In [180…
```
sbn.barplot(x='the_total_mean', y='year', data=d7, orient='h', ci='sd')
#wersja z std dev
```

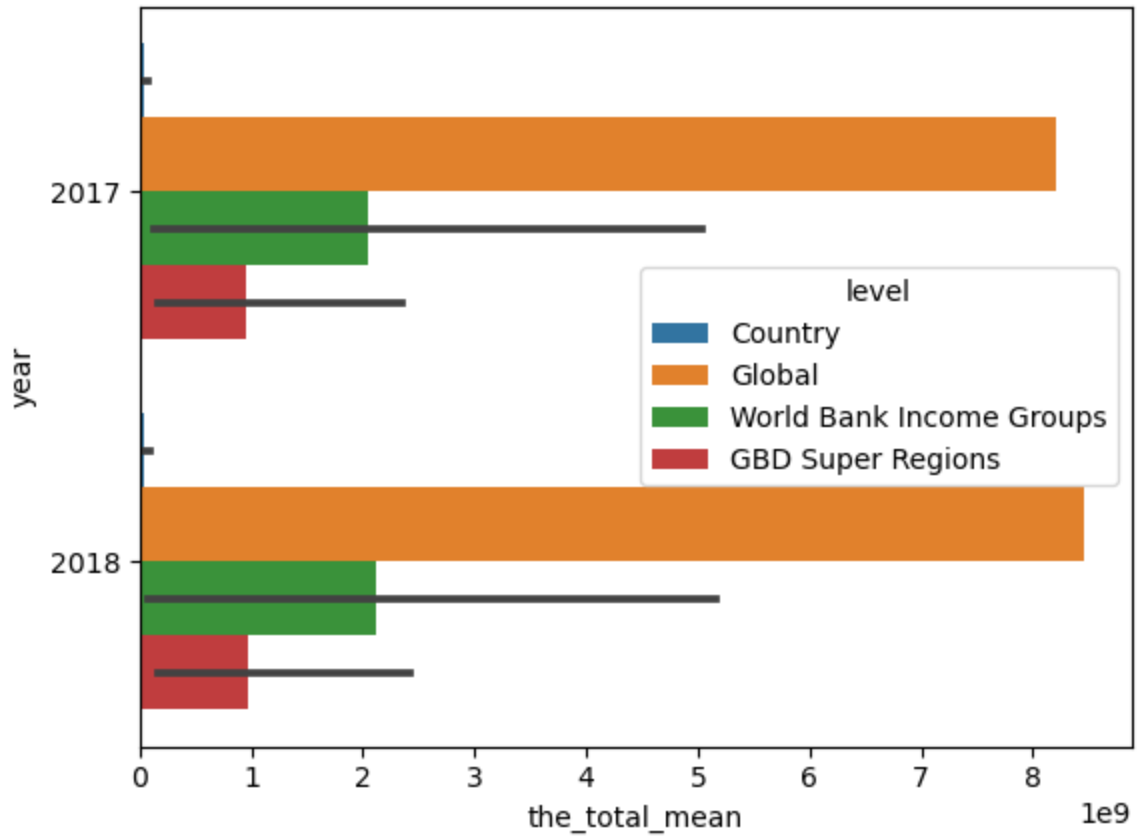
Out[180]:
```
<AxesSubplot:xlabel='the_total_mean', ylabel='year'>
```




Dodanie nowej wartości kategorycznej hue

In [182…
```
d7 = d7[d7["year"] > 2016]
sbn.barplot(x='the_total_mean', y='year', hue='level', data=d7, orient='h')
```

Out[182]: <AxesSubplot:xlabel='the_total_mean', ylabel='year'>



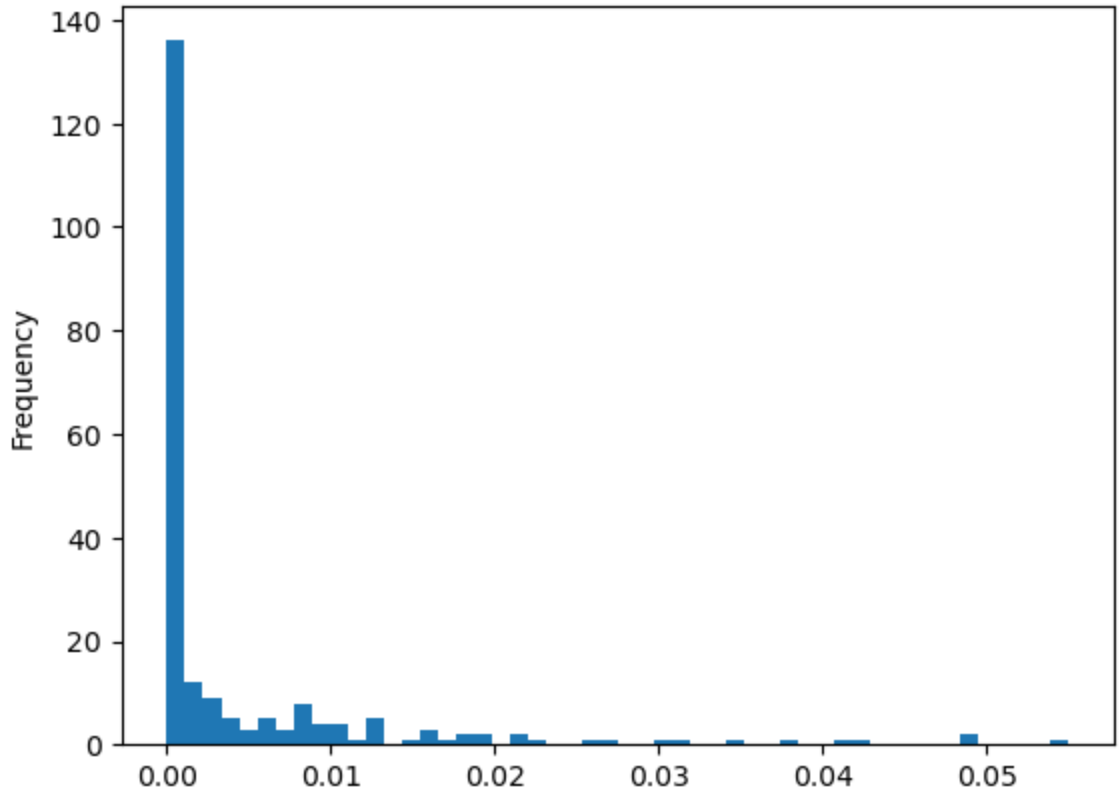

Histogram

In [183…
```
d8 = df[df["year"] > 2017]
d8 = d8["dah_per_gdp_mean"]
d8
```

Out[183]:
```
23      0.013
47      0.001
71      0.000
95      0.000
119     0.000
        ...
5135    0.000
5182    0.001
5183    0.001
5207    0.000
5231    0.006
Name: dah_per_gdp_mean, Length: 218, dtype: float64
```

In [184…
```
d8.plot.hist(bins=50)
```

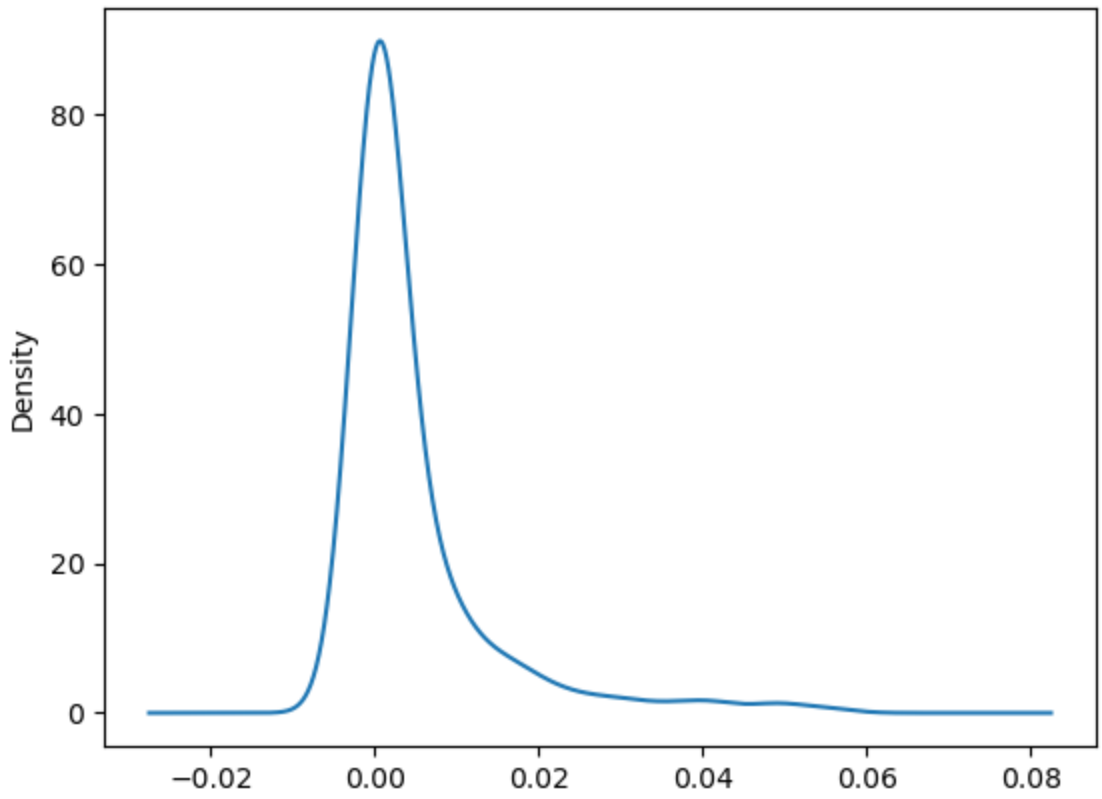
Out[184]: <AxesSubplot:ylabel='Frequency'>




Wykres gęstości - wykres prawdopodbieństwa obserwowanych danych

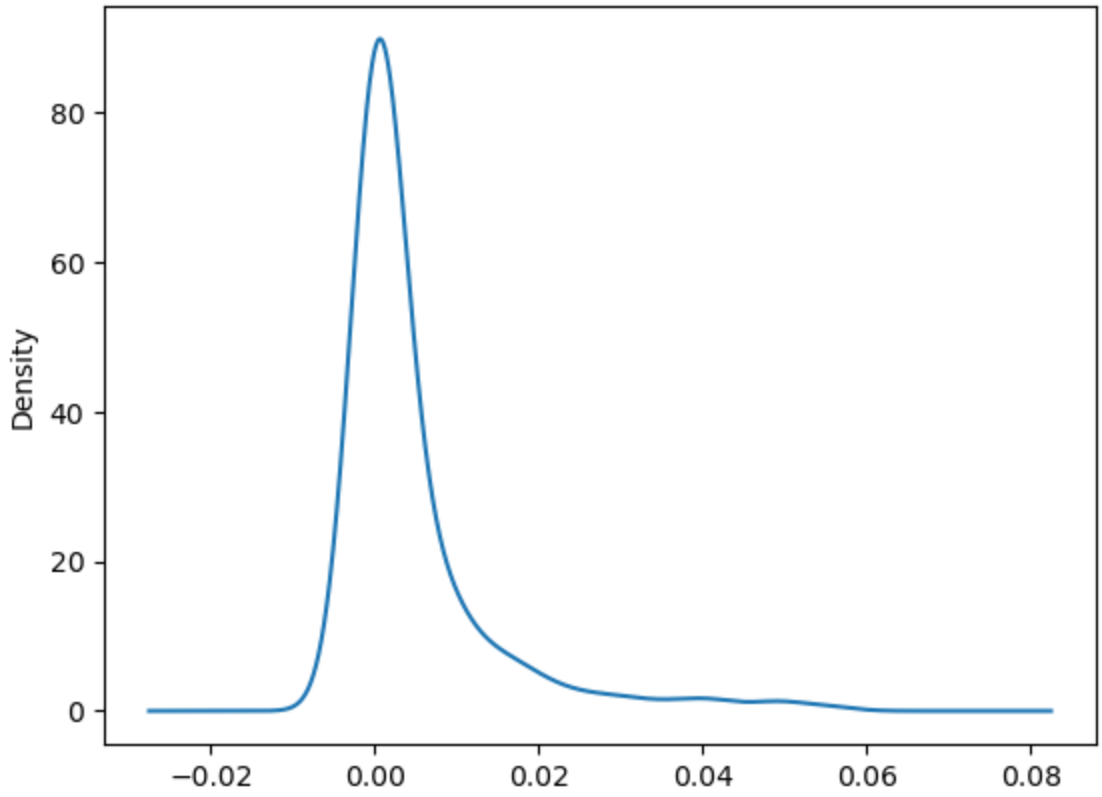
In [185…
```
d8.plot.kde()
```

Out[185]:
```
<AxesSubplot:ylabel='Density'>
```



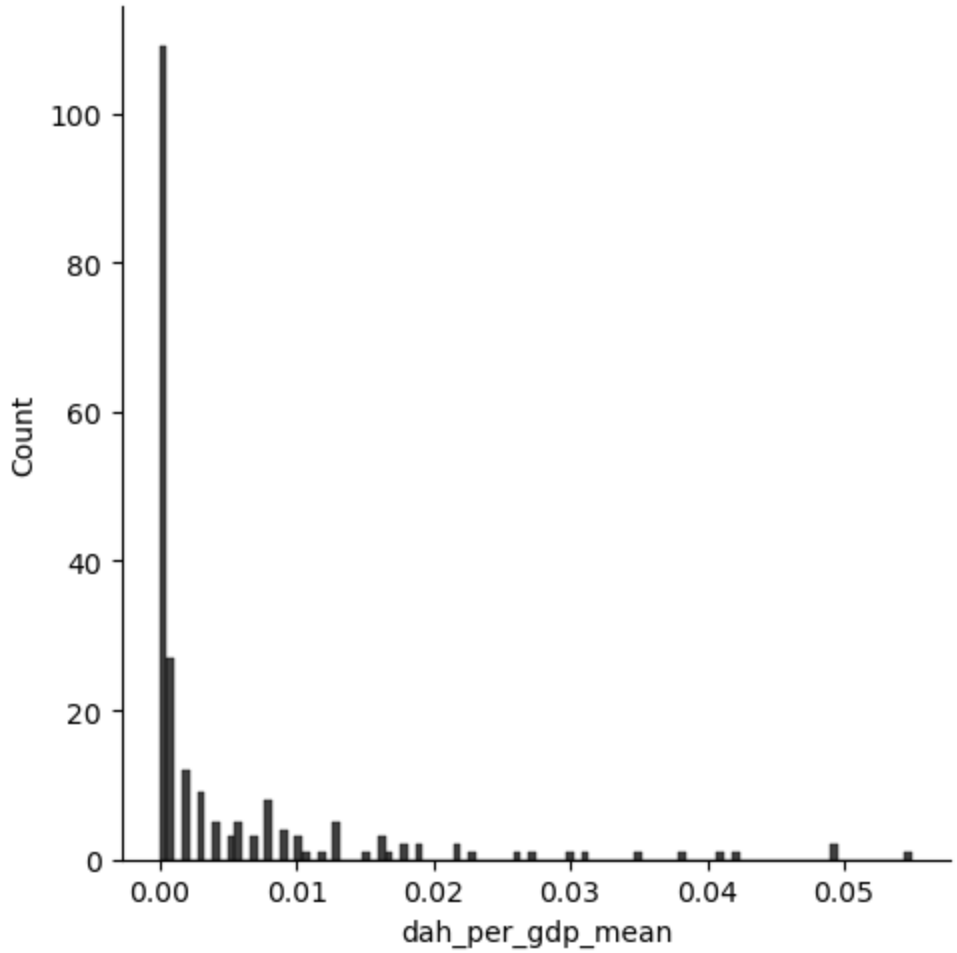

In [186…
```
d8.plot.density()
```

Out[186]:
```
<AxesSubplot:ylabel='Density'>
```



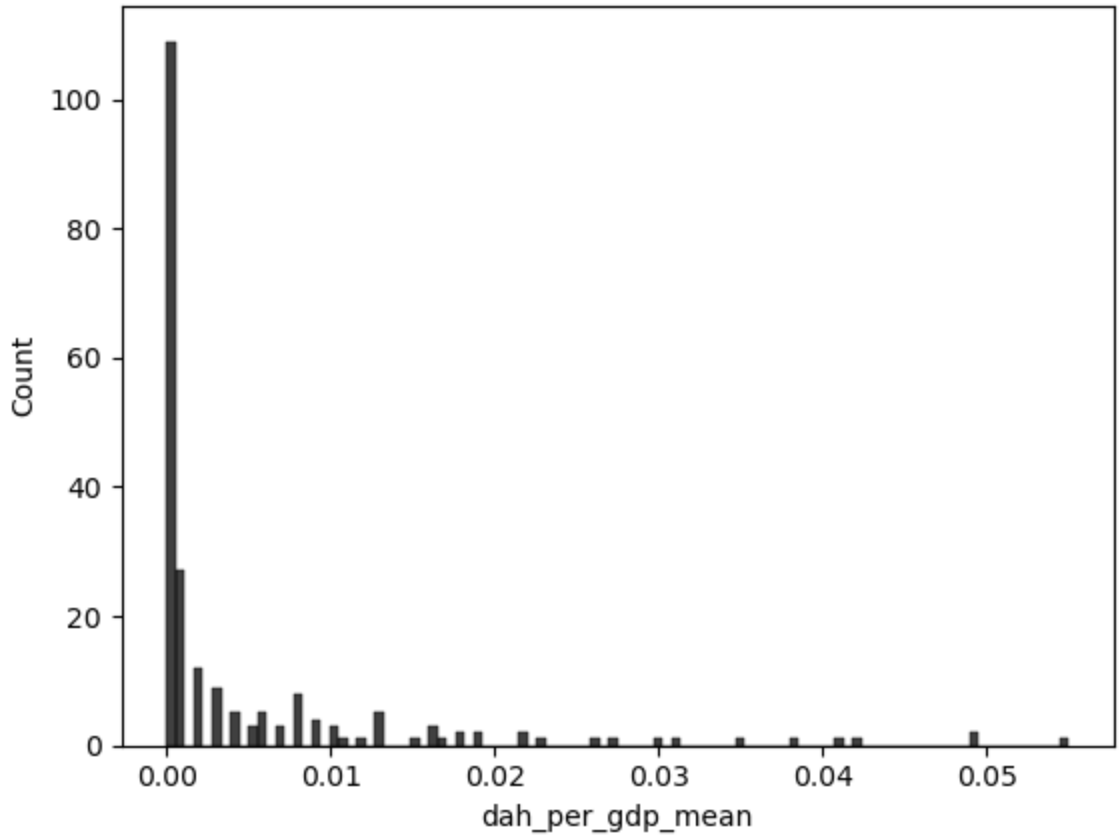

In [187…
```
sbn.displot(d8, bins=100, color='k')
```

Out[187]:
```
<seaborn.axisgrid.FacetGrid at 0x16498f7d820>
```




In [188…
```
sbn.histplot(d8, bins=100, color='k')
```

Out[188]:
```
<AxesSubplot:xlabel='dah_per_gdp_mean', ylabel='Count'>
```




Wykresy punktowe i bitowe

In [189…

```
d9 = df
d9 = d9[d9["year"] > 2012]
d9
```

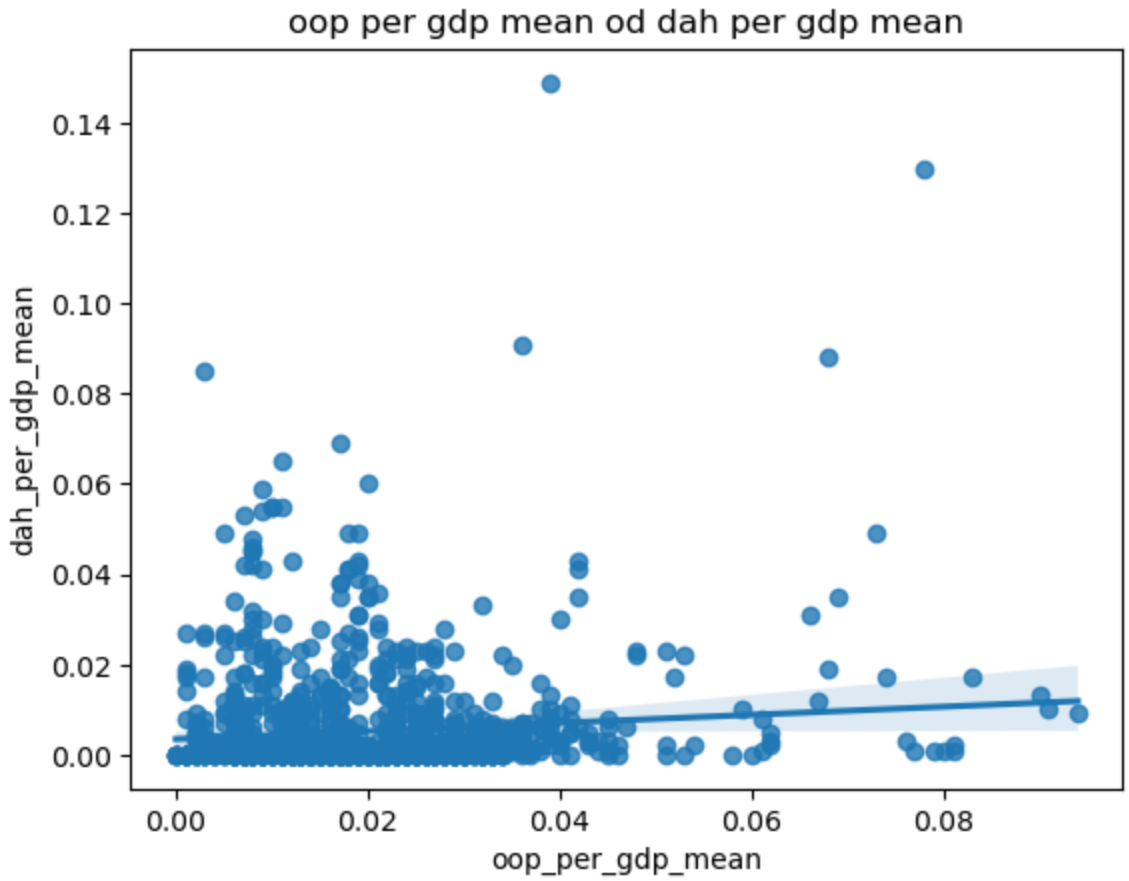
Out[189]:

| | location_id | location_name | iso3 | level | year | the_total_mean | the_total_lower | the_total_upper | the_total |
|---|---|---|---|---|---|---|---|---|---|
| **18** | 160 | Afghanistan | AFG | Country | 2013 | 1554494 | 1435933 | 1674751 | |
| **19** | 160 | Afghanistan | AFG | Country | 2014 | 1664981 | 1541417 | 1800126 | |
| **20** | 160 | Afghanistan | AFG | Country | 2015 | 1822293 | 1693212 | 1960564 | |
| **21** | 160 | Afghanistan | AFG | Country | 2016 | 1882415 | 1747511 | 2028013 | |
| **22** | 160 | Afghanistan | AFG | Country | 2017 | 1992736 | 1857448 | 2142606 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **5227** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2014 | 71349683 | 69713253 | 73018691 | |
| **5228** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2015 | 73465621 | 71828527 | 75186691 | |
| **5229** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2016 | 73972108 | 72267558 | 75733371 | |
| **5230** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2017 | 77406545 | 75367505 | 79506397 | |
| **5231** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2018 | 78994327 | 76466329 | 81632889 | |

1308 rows × 84 columns

In [190…
```
sbn.regplot(x='oop_per_gdp_mean', y='dah_per_gdp_mean', data=d9)
plt.title('oop per gdp mean od dah per gdp mean'.format('oop_per_gdp_mean', 'dah_per_gdp
```

Out[190]:
```
Text(0.5, 1.0, 'oop per gdp mean od dah per gdp mean')
```




Macierz wykresów

In [191…
```
d10 = df
d10 = d10[d10["year"] > 2010].head(50)
d10 = d10[["the_total_mean", "the_total_ppp_mean", "dah_per_gdp_mean", "oop_per_gdp_mean
d10
```

Out[191]:

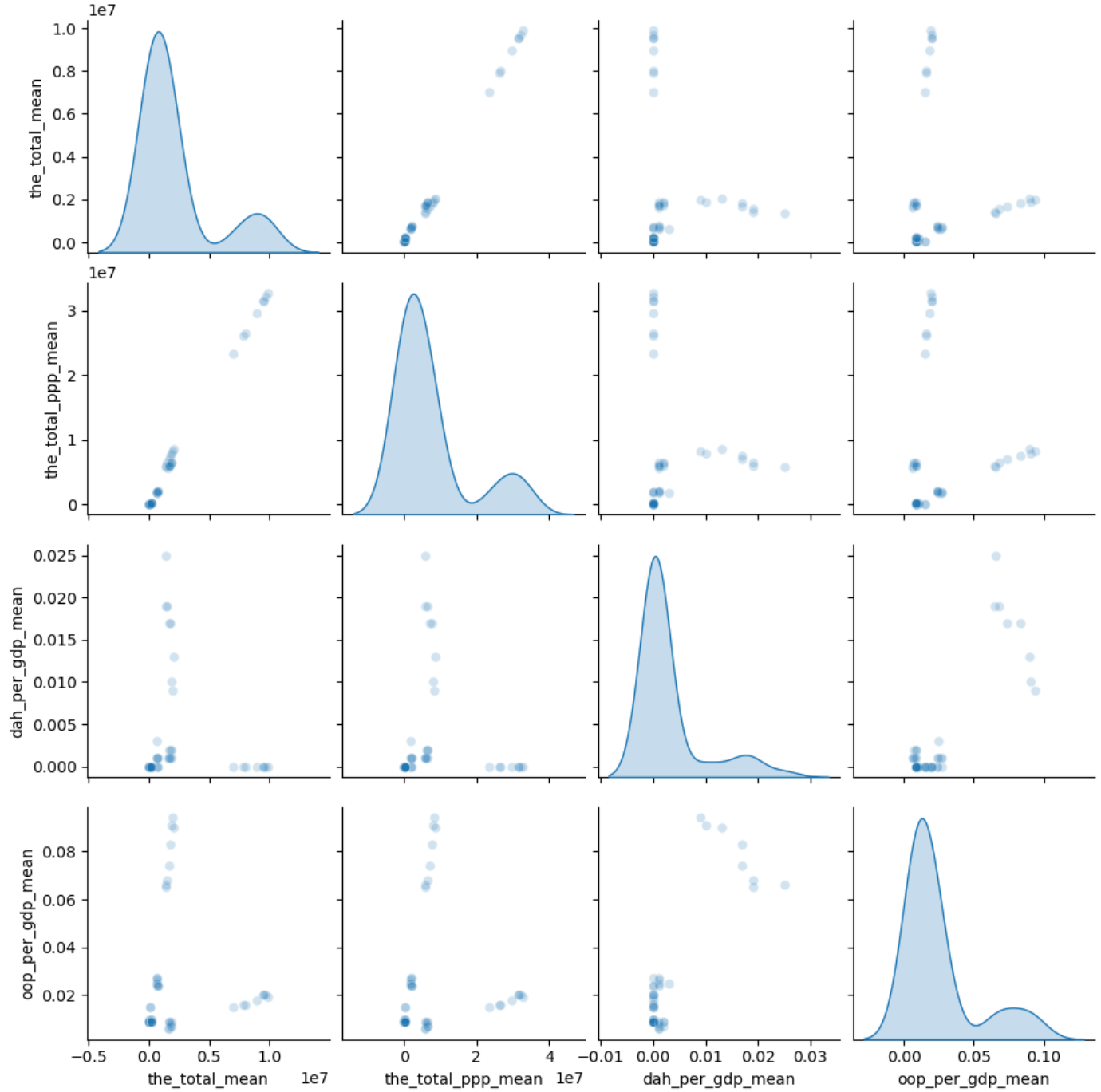
| | the_total_mean | the_total_ppp_mean | dah_per_gdp_mean | oop_per_gdp_mean |
|---|---|---|---|---|
| **16** | 1373092 | 5699060 | 0.025 | 0.066 |
| **17** | 1410474 | 5854216 | 0.019 | 0.065 |
| **18** | 1554494 | 6451977 | 0.019 | 0.068 |
| **19** | 1664981 | 6910557 | 0.017 | 0.074 |
| **20** | 1822293 | 7563486 | 0.017 | 0.083 |
| **21** | 1882415 | 7813023 | 0.010 | 0.091 |
| **22** | 1992736 | 8270915 | 0.009 | 0.094 |
| **23** | 2037487 | 8456655 | 0.013 | 0.090 |
| **40** | 639655 | 1782716 | 0.003 | 0.025 |
| **41** | 641496 | 1787846 | 0.001 | 0.026 |
| **42** | 659893 | 1839119 | 0.000 | 0.027 |

| | | | | |
|---|---|---|---|---|
| **43** | 707623 | 1972142 | 0.001 | 0.027 |
| **44** | 689829 | 1922550 | 0.001 | 0.025 |
| **45** | 690713 | 1925014 | 0.000 | 0.024 |
| **46** | 724906 | 2020310 | 0.000 | 0.024 |
| **47** | 773638 | 2156125 | 0.001 | 0.024 |
| **64** | 7035126 | 23317461 | 0.000 | 0.015 |
| **65** | 8003953 | 26528574 | 0.000 | 0.016 |
| **66** | 7880135 | 26118187 | 0.000 | 0.016 |
| **67** | 8926232 | 29585404 | 0.000 | 0.018 |
| **68** | 9866905 | 32703203 | 0.000 | 0.019 |
| **69** | 9690605 | 32118869 | 0.000 | 0.020 |
| **70** | 9528786 | 31582532 | 0.000 | 0.020 |
| **71** | 9502698 | 31496063 | 0.000 | 0.020 |
| **88** | 33761 | 33761 | 0.000 | 0.009 |
| **89** | 32557 | 32557 | 0.000 | 0.009 |
| **90** | 31775 | 31775 | 0.000 | 0.009 |
| **91** | 31895 | 31895 | 0.000 | 0.009 |
| **92** | 32106 | 32106 | 0.000 | 0.009 |
| **93** | 31926 | 31926 | 0.000 | 0.009 |
| **94** | 31253 | 31253 | 0.000 | 0.010 |
| **95** | 32291 | 32291 | 0.000 | 0.009 |
| **112** | 204699 | 218073 | 0.000 | 0.009 |
| **113** | 197328 | 210220 | 0.000 | 0.010 |
| **114** | 196817 | 209676 | 0.000 | 0.010 |
| **115** | 197054 | 209929 | 0.000 | 0.009 |
| **116** | 201031 | 214165 | 0.000 | 0.009 |
| **117** | 216149 | 230271 | 0.000 | 0.009 |
| **118** | 225647 | 240390 | 0.000 | 0.009 |
| **119** | 234628 | 249957 | 0.000 | 0.009 |
| **136** | 1642328 | 5670017 | 0.001 | 0.006 |
| **137** | 1705780 | 5889079 | 0.001 | 0.006 |
| **138** | 1823689 | 6296151 | 0.002 | 0.007 |
| **139** | 1854933 | 6404019 | 0.001 | 0.007 |
| **140** | 1761334 | 6080878 | 0.001 | 0.008 |
| **141** | 1740257 | 6008108 | 0.002 | 0.009 |
| **142** | 1857264 | 6412066 | 0.002 | 0.009 |
| **143** | 1707322 | 5894405 | 0.001 | 0.009 |
| **160** | 69838 | 92019 | 0.000 | 0.015 |

| | | | | |
|---|---|---|---|---|
| **161** | 69674 | 91803 | 0.000 | 0.015 |

In [192…
```
sbn.pairplot(d10, diag_kind='kde', plot_kws={'alpha': 0.2})
```

Out[192]:
```
<seaborn.axisgrid.PairGrid at 0x1649a43d160>
```




Dane kategoryczne

In [193…
```
d11 = df
d11 = d11[(d11["iso3"] == "S3") | (d11["iso3"] == "USA")  | (d11["iso3"] == "CHN")]
d11 = d11[d11["year"] > 2010]
d11
```

Out[193]:

| | location_id | location_name | iso3 | level | year | the_total_mean | the_total_lower | the_total_upper | the_total |
|---|---|---|---|---|---|---|---|---|---|
| **904** | 6 | China | CHN | Country | 2011 | 394499221 | 363955067 | 425030458 | |
| **905** | 6 | China | CHN | Country | 2012 | 438607436 | 408300708 | 470912636 | |
| **906** | 6 | China | CHN | Country | 2013 | 481628318 | 451394063 | 515950870 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **907** | 6 | China | CHN | Country | 2014 | 525586094 | 494048109 | 558205692 |
| **908** | 6 | China | CHN | Country | 2015 | 584192485 | 551263247 | 618708603 |
| **909** | 6 | China | CHN | Country | 2016 | 632511907 | 600721292 | 665591927 |
| **910** | 6 | China | CHN | Country | 2017 | 677343796 | 644644563 | 710047114 |
| **911** | 6 | China | CHN | Country | 2018 | 738123387 | 699261899 | 777328486 |
| **4696** | 102 | United States of America | USA | Country | 2011 | 2944162745 | 2890250905 | 2997085936 |
| **4697** | 102 | United States of America | USA | Country | 2012 | 2997075946 | 2948522176 | 3051844586 |
| **4698** | 102 | United States of America | USA | Country | 2013 | 3006573573 | 2953445463 | 3058500115 |
| **4699** | 102 | United States of America | USA | Country | 2014 | 3209771501 | 3159094694 | 3265764647 |
| **4700** | 102 | United States of America | USA | Country | 2015 | 3303976697 | 3250922048 | 3357576999 |
| **4701** | 102 | United States of America | USA | Country | 2016 | 3423031116 | 3370628273 | 3471875028 |
| **4702** | 102 | United States of America | USA | Country | 2017 | 3502754273 | 3448933896 | 3556604395 |
| **4703** | 102 | United States of America | USA | Country | 2018 | 3598369743 | 3529855473 | 3668158130 |
| **5224** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2011 | 62011236 | 60507521 | 63569807 |
| **5225** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2012 | 65111564 | 63505530 | 66755373 |
| **5226** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2013 | 68804826 | 67166034 | 70485588 |
| **5227** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2014 | 71349683 | 69713253 | 73018691 |
| **5228** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2015 | 73465621 | 71828527 | 75186691 |
| **5229** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2016 | 73972108 | 72267558 | 75733371 |
| **5230** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2017 | 77406545 | 75367505 | 79506397 |
| **5231** | 166 | Sub-Saharan Africa | S3 | GBD Super Regions | 2018 | 78994327 | 76466329 | 81632889 |

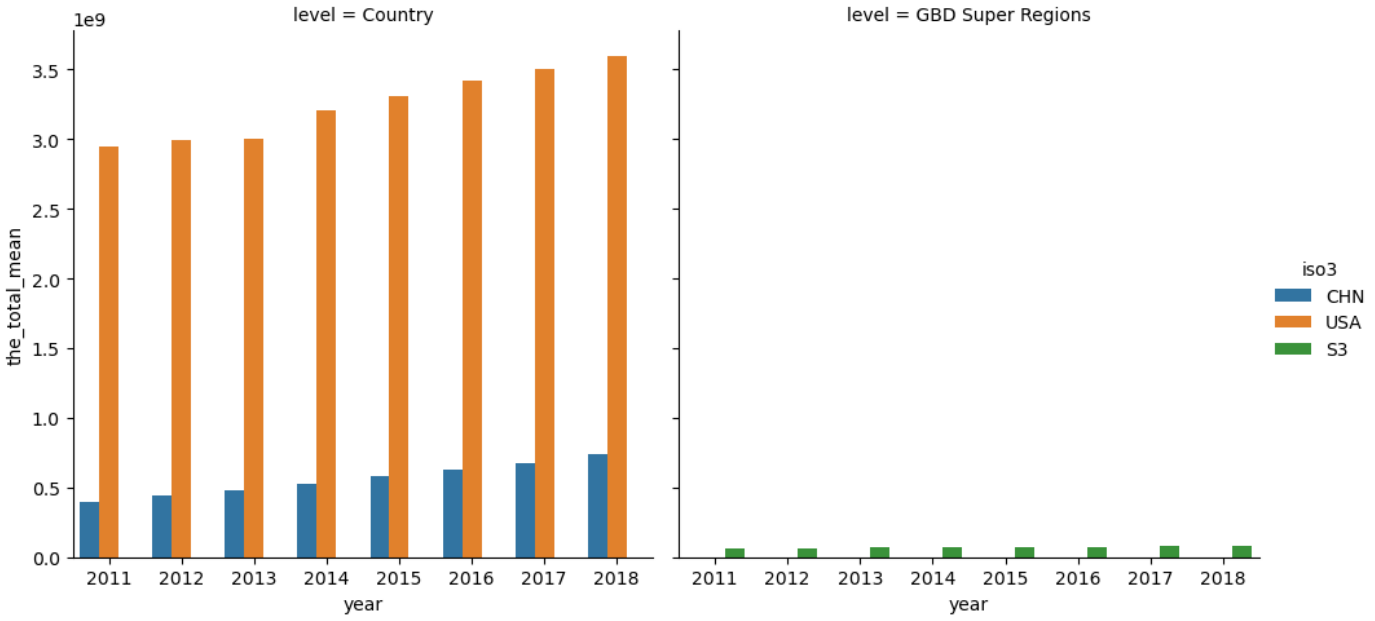
24 rows × 84 columns

In [194…

```
sbn.catplot(x='year', y='the_total_mean', hue='iso3', col='level', kind='bar', data=d11)
```

```
#danymi kategoryzującymi jest level
```

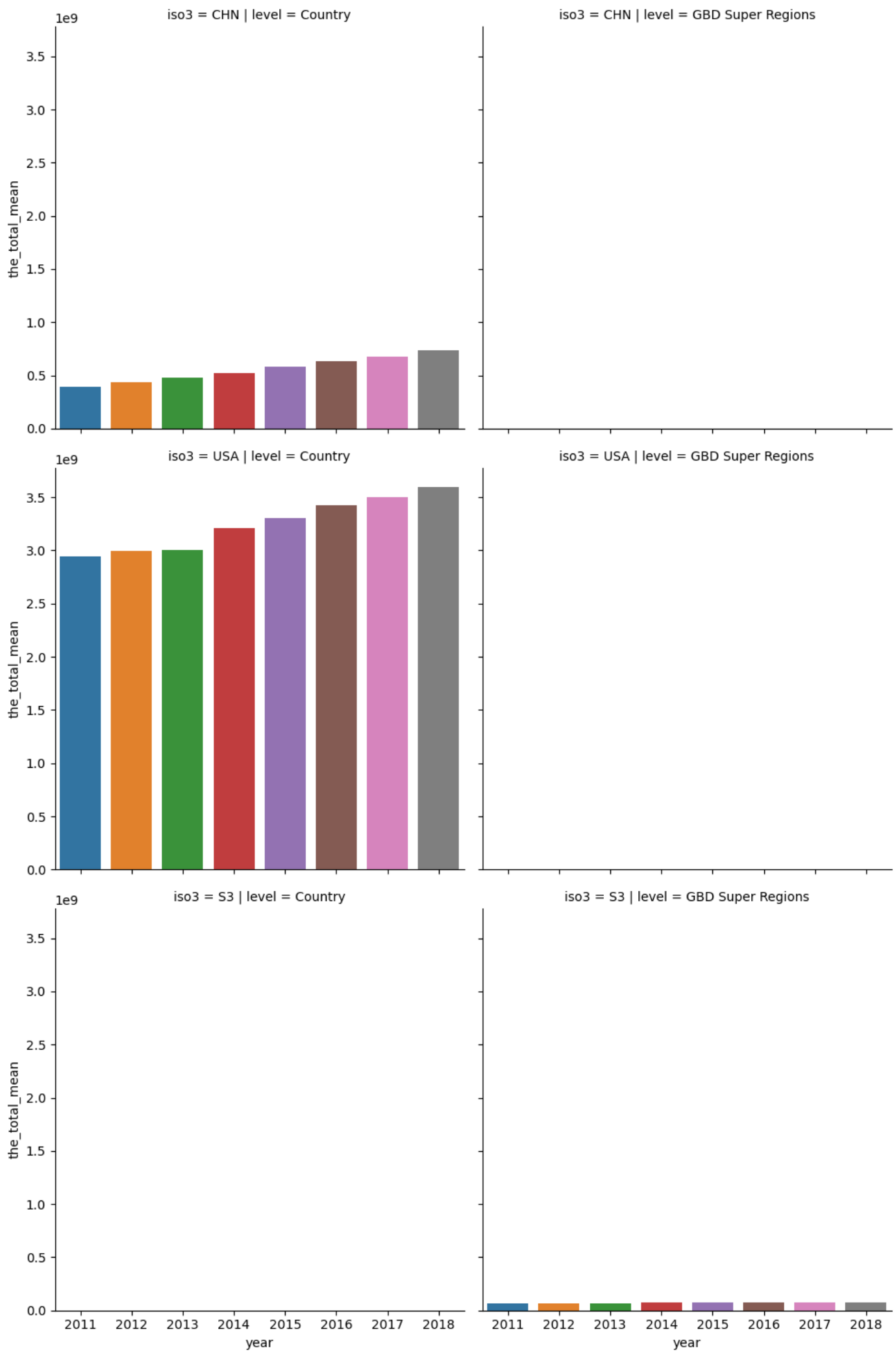
Out[194]:
```
<seaborn.axisgrid.FacetGrid at 0x1649a2a6460>
```




Siatka aspektów

In [207…
```
sbn.catplot(x='year', y='the_total_mean', row='iso3', col='level', kind='bar', data=d11)
```

Out[207]:
```
<seaborn.axisgrid.FacetGrid at 0x1649b76e640>
```


iso3 = CHN | level = Country
iso3 = CHN | level = GBD Super Regions
iso3 = USA | level = Country
iso3 = USA | level = GBD Super Regions
iso3 = S3 | level = Country
iso3 = S3 | level = GBD Super Regions
1e9
the_total_mean
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
2011
2012
2013
2014
2015
2016
2017
2018
year

Wykresy blokowe

In [249…
```
d12 = df[df["year"] > 2014]
d12 = d12[(d12["location_name"] == "Albania") | (d12["location_name"] == "Egypt") |
          (d12["location_name"] == "Latvia") | (d12["location_name"] == "Netherlands")]
d12
```

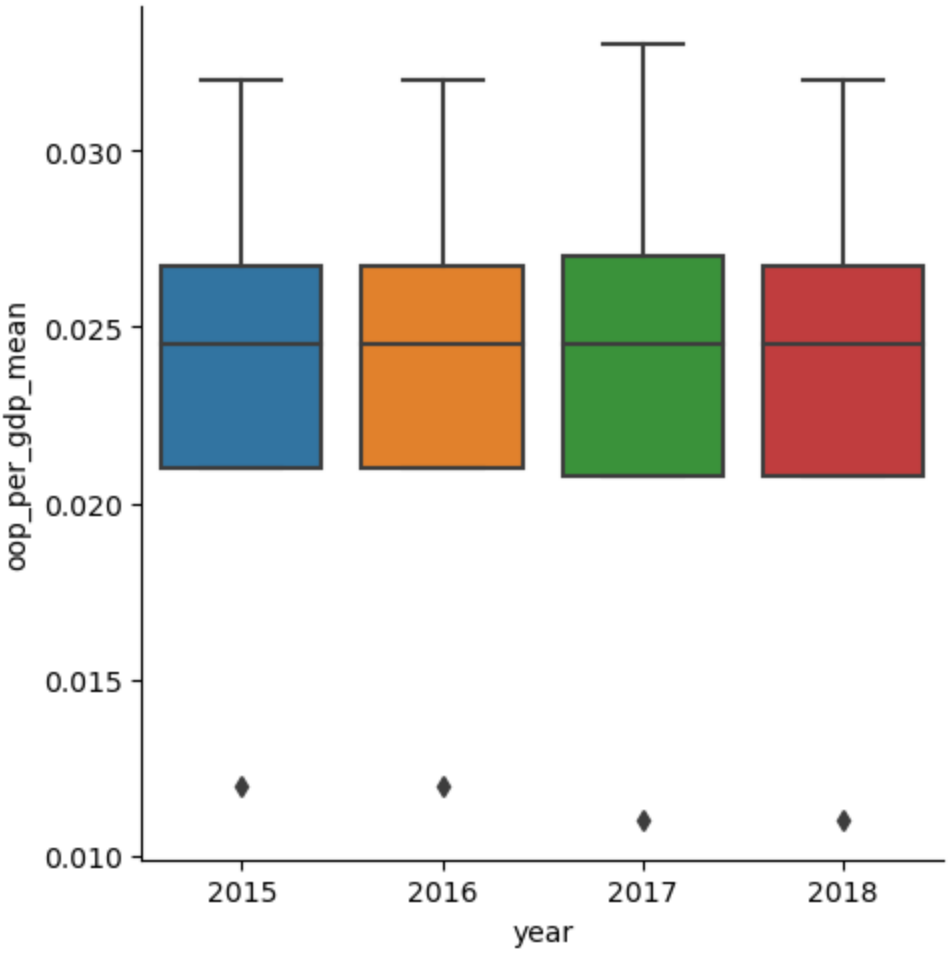
Out[249]:

| | location_id | location_name | iso3 | level | year | the_total_mean | the_total_lower | the_total_upper | the_total |
|---|---|---|---|---|---|---|---|---|---|
| **44** | 43 | Albania | ALB | Country | 2015 | 689829 | 646378 | 738745 | |
| **45** | 43 | Albania | ALB | Country | 2016 | 690713 | 644271 | 738868 | |
| **46** | 43 | Albania | ALB | Country | 2017 | 724906 | 671771 | 779525 | |
| **47** | 43 | Albania | ALB | Country | 2018 | 773638 | 714327 | 838110 | |
| **1340** | 141 | Egypt | EGY | Country | 2015 | 15071072 | 14381859 | 15742578 | |
| **1341** | 141 | Egypt | EGY | Country | 2016 | 16078051 | 15343614 | 16813419 | |
| **1342** | 141 | Egypt | EGY | Country | 2017 | 16881770 | 16125703 | 17698948 | |
| **1343** | 141 | Egypt | EGY | Country | 2018 | 16764374 | 15786043 | 17787561 | |
| **2372** | 59 | Latvia | LVA | Country | 2015 | 1801007 | 1753510 | 1853748 | |
| **2373** | 59 | Latvia | LVA | Country | 2016 | 1897400 | 1847688 | 1946688 | |
| **2374** | 59 | Latvia | LVA | Country | 2017 | 2009249 | 1960480 | 2060583 | |
| **2375** | 59 | Latvia | LVA | Country | 2018 | 2131696 | 2066178 | 2198799 | |
| **3020** | 89 | Netherlands | NLD | Country | 2015 | 88930899 | 87386562 | 90536069 | |
| **3021** | 89 | Netherlands | NLD | Country | 2016 | 89554408 | 87964857 | 91154289 | |
| **3022** | 89 | Netherlands | NLD | Country | 2017 | 90480130 | 88820947 | 92135207 | |
| **3023** | 89 | Netherlands | NLD | Country | 2018 | 92143985 | 90088095 | 94254403 | |

16 rows × 84 columns

In [250…
```
sbn.catplot(x='year', y='oop_per_gdp_mean', kind='box', data=d12)
#mediana, kwartyle, wartości odstające
```

Out[250]:
```
<seaborn.axisgrid.FacetGrid at 0x164a5988cd0>
```


0.030
0.025
0.020
0.015
0.010
oop_per_gdp_mean
2015
2016
2017
2018
year