In [1]:
```
# 1. Wczytywanie danych i wyświetlanie podstawowych informacji
import pandas as pd
df = pd.read_csv('IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv')
print(df.head())
print(df.info())
print(df.describe())
```

```
   location_id             location_name  year  sex_id   sex  age_group_id  \
0          102  United States of America  2016       1  Male           158   
1          102  United States of America  2016       1  Male           170   
2          102  United States of America  2016       1  Male           171   
3          102  United States of America  2016       1  Male           154   
4          102  United States of America  2016       1  Male            22   

  age_group_name acause  cause_name  risk_id  \
0      <20 years   _all  All causes       82   
1       20 to 44   _all  All causes       82   
2       45 to 64   _all  All causes       82   
3        65 plus   _all  All causes       82   
4       All Ages   _all  All causes       82   

                                 risk_name          metric        mean  \
0  Unsafe water, sanitation, and handwashing  2016 US Dollar   52.810750   
1  Unsafe water, sanitation, and handwashing  2016 US Dollar   48.000244   
2  Unsafe water, sanitation, and handwashing  2016 US Dollar   69.765933   
3  Unsafe water, sanitation, and handwashing  2016 US Dollar   88.960404   
4  Unsafe water, sanitation, and handwashing  2016 US Dollar  259.537331   

        lower       upper  
0   34.776353   76.466271  
1   31.644490   70.893323  
2   45.764417  101.797946  
3   59.168652  128.091201  
4  171.226033  375.015989  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1241 entries, 0 to 1240
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   location_id     1241 non-null   int64  
 1   location_name   1241 non-null   object 
 2   year            1241 non-null   int64  
 3   sex_id          1241 non-null   int64  
 4   sex             1241 non-null   object 
 5   age_group_id    1241 non-null   int64  
 6   age_group_name  1241 non-null   object 
 7   acause          1241 non-null   object 
 8   cause_name      1241 non-null   object 
 9   risk_id         1241 non-null   int64  
 10  risk_name       1241 non-null   object 
 11  metric          1241 non-null   object 
 12  mean            1241 non-null   float64
 13  lower           1241 non-null   float64
 14  upper           1241 non-null   float64
dtypes: float64(3), int64(5), object(7)
memory usage: 145.6+ KB
None
       location_id    year       sex_id  age_group_id      risk_id  \
count       1241.0  1241.0  1241.000000   1241.000000  1241.000000   
mean         102.0  2016.0     2.020145    132.887188   129.024174   
std            0.0     0.0     0.812452     59.017044    73.481341   
min          102.0  2016.0     1.000000     22.000000    82.000000   
25%          102.0  2016.0     1.000000    154.000000    98.000000   
50%          102.0  2016.0     2.000000    158.000000   105.000000   
75%          102.0  2016.0     3.000000    170.000000   125.000000   
max          102.0  2016.0     3.000000    171.000000   381.000000   
```

```
                mean          lower          upper
count    1241.000000    1241.000000    1241.000000
mean     7523.642696    5972.186569    9227.098980
std     18443.451439   15681.094146   21316.739752
min         0.058442   -4401.524592       0.105215
25%       275.378150     129.072794     421.882542
50%      1372.476758     821.243571    2097.737090
75%      5763.893904    4033.397743    7515.218944
max    238503.405500  178217.189400  291573.753200
```

In [3]:

```python
# 2. Obliczanie podstawowych statystyk
mean_lower = df['lower'].mean()
print('Średnia wartość lower:', mean_lower)
mean_upper = df['upper'].mean()
print('Średnia wartość upper:', mean_upper)
median_lower = df['lower'].median()
print('Mediana_lower:', median_lower)
median_upper = df['upper'].median()
print('Mediana pper:', median_upper)
std_lower = df['lower'].std()
print('Odchylenie standardowe wartości lower:', std_lower)
std_upper = df['upper'].std()
print('Odchylenie standardowe wartości upper:', std_upper)
```

```
Średnia wartość lower: 5972.186569134251
Średnia wartość upper: 9227.098980246341
Mediana_lower: 821.2435711
Mediana pper: 2097.73709
Odchylenie standardowe wartości lower: 15681.094145702247
Odchylenie standardowe wartości upper: 21316.739751631947
```

In [4]:

```python
# 3. Identyfikacja i obsluga brakujących danych
missing_values = df.isnull().sum()
print('Liczba brakujących wartości w poszczególnych kolumnach:')
print(missing_values)
```

```
Liczba brakujących wartości w poszczególnych kolumnach:
location_id       0
location_name     0
year              0
sex_id            0
sex               0
age_group_id      0
age_group_name    0
acause            0
cause_name        0
risk_id           0
risk_name         0
metric            0
mean              0
lower             0
upper             0
dtype: int64
```

In [6]:
```
# 3. Identyfikacja i obsluga brakujących danych (c.d.)

# Uzupełnianie brakujących wartości iso3 (kod kraju) na podstawie location_name
df['location_id'] = df['location_id'].fillna(df['location_name'].str[:3])
missing_values = df.isnull().sum()
print('Liczba brakujących wartości w poszczególnych kolumnach po uzupełnieniu:')
print(missing_values)
```

```
Liczba brakujących wartości w poszczególnych kolumnach po uzupełnieniu:
location_id       0
location_name     0
year              0
sex_id            0
sex               0
age_group_id      0
age_group_name    0
acause            0
cause_name        0
risk_id           0
risk_name         0
metric            0
mean              0
lower             0
upper             0
dtype: int64
```

In [7]:
```
# 4. Wykrywanie wartości odstających (używając metody IRQ):
Q1 = df['upper'].quantile(0.25)
Q3 = df['upper'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df['upper'] < lower_bound) | (df['upper'] > upper_bound)]
print('Wartości odstające w kolumnie upper:')
print(outliers)
```

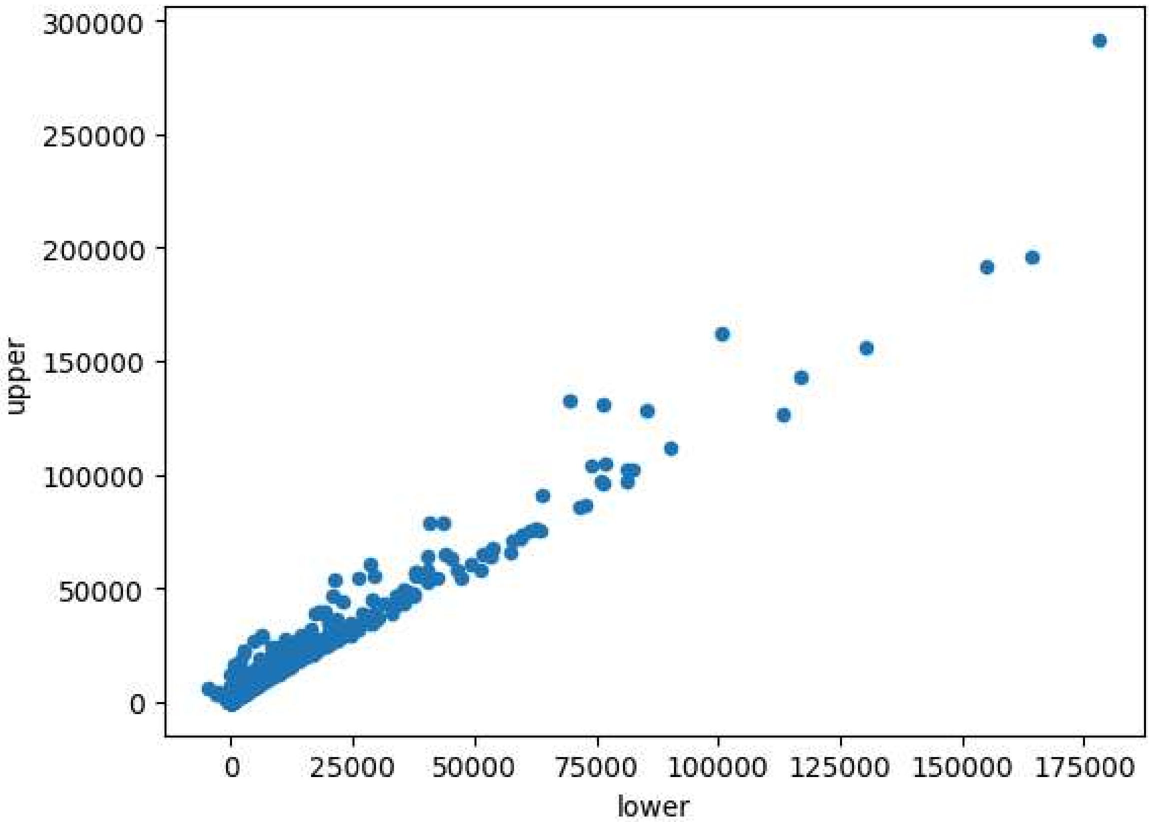
```
Wartości odstające w kolumnie upper:
      location_id              location_name  year  sex_id     sex  \
19            102  United States of America  2016       1    Male   
24            102  United States of America  2016       2  Female   
27            102  United States of America  2016       3    Both   
28            102  United States of America  2016       3    Both   
29            102  United States of America  2016       3    Both   
...           ...                       ...   ...     ...     ...   
1219          102  United States of America  2016       2  Female   
1220          102  United States of America  2016       2  Female   
1223          102  United States of America  2016       3    Both   
1224          102  United States of America  2016       3    Both   
1225          102  United States of America  2016       3    Both   

      age_group_id age_group_name acause                   cause_name  \
19              22       All Ages   _all                   All causes   
24              22       All Ages   _all                   All causes   
27             171       45 to 64   _all                   All causes   
28             154        65 plus   _all                   All causes   
29              22       All Ages   _all                   All causes   
...            ...            ...    ...                          ...   
1219           154        65 plus     rf  Expenditure on risk factors   
1220            22       All Ages     rf  Expenditure on risk factors   
1223           171       45 to 64     rf  Expenditure on risk factors   
1224           154        65 plus     rf  Expenditure on risk factors   
1225            22       All Ages     rf  Expenditure on risk factors   

      risk_id                    risk_name          metric         mean  \
19         85                Air pollution  2016 US Dollar  16732.36082   
24         85                Air pollution  2016 US Dollar  16866.90331   
27         85                Air pollution  2016 US Dollar  14330.36646   
28         85                Air pollution  2016 US Dollar  16791.27025   
29         85                Air pollution  2016 US Dollar  33599.26413   
...       ...                          ...             ...          ...   
1219      107  High systolic blood pressure  2016 US Dollar  22473.84815   
1220      107  High systolic blood pressure  2016 US Dollar  41571.35674   
1223      107  High systolic blood pressure  2016 US Dollar  31838.85583   
1224      107  High systolic blood pressure  2016 US Dollar  40453.92810   
1225      107  High systolic blood pressure  2016 US Dollar  78978.20846   

             lower        upper  
19    10715.488990  23388.85046  
24    10267.186300  23763.72018  
27     8466.798715  20733.65741  
28    10489.090960  23137.31274  
29    21011.760530  46983.89362  
...            ...          ...  
1219  20029.878090  26384.52697  
1220  37609.392360  48057.33818  
1223  28621.213930  34994.85259  
1224  35717.040250  45794.07061  
1225  72635.368470  86818.86422  

[166 rows x 15 columns]
```

In [8]:

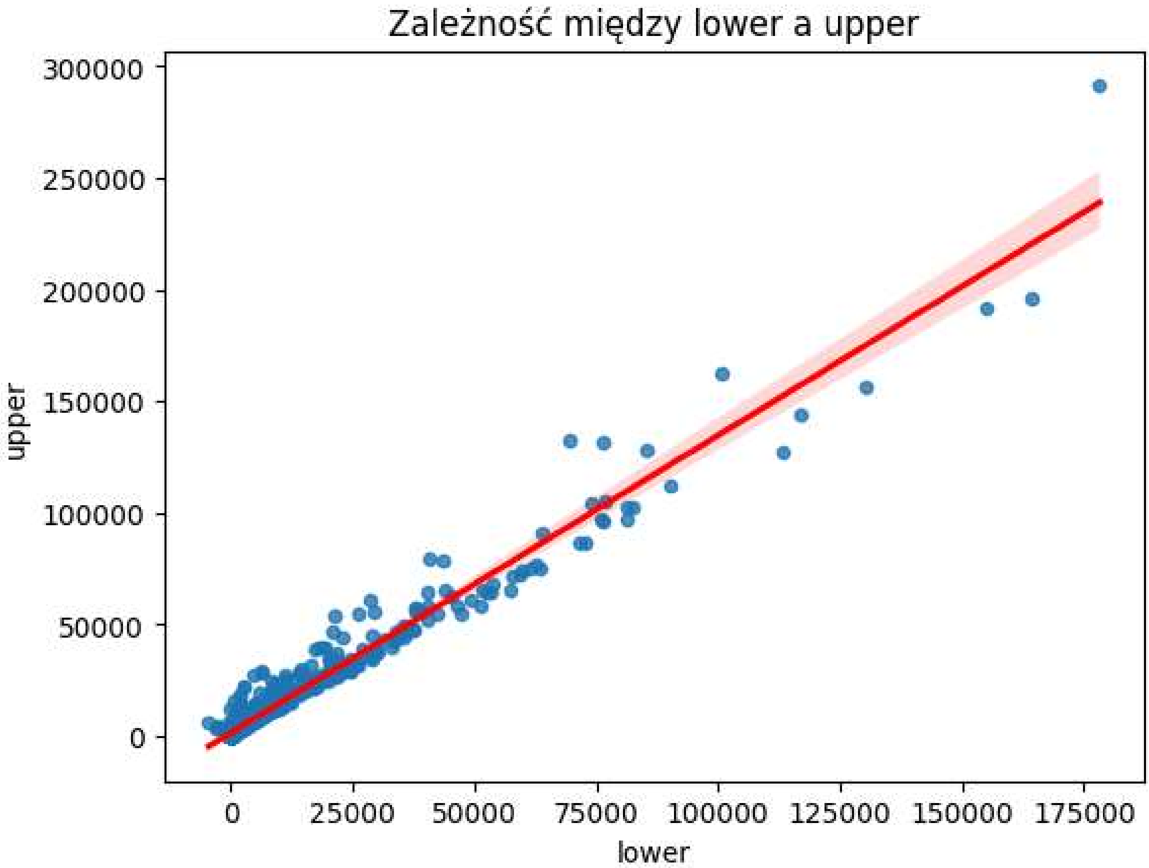
```python
# 5. Analiza zależności między kolumnami
numeric_df = df.select_dtypes(include=['number']) # wybierz tylko kolumny numery
correlation = numeric_df.corr()
# print('Macierz korelacji:')
```

```
# print(correlation)
numeric_df.plot.scatter(x='lower', y='upper')
```

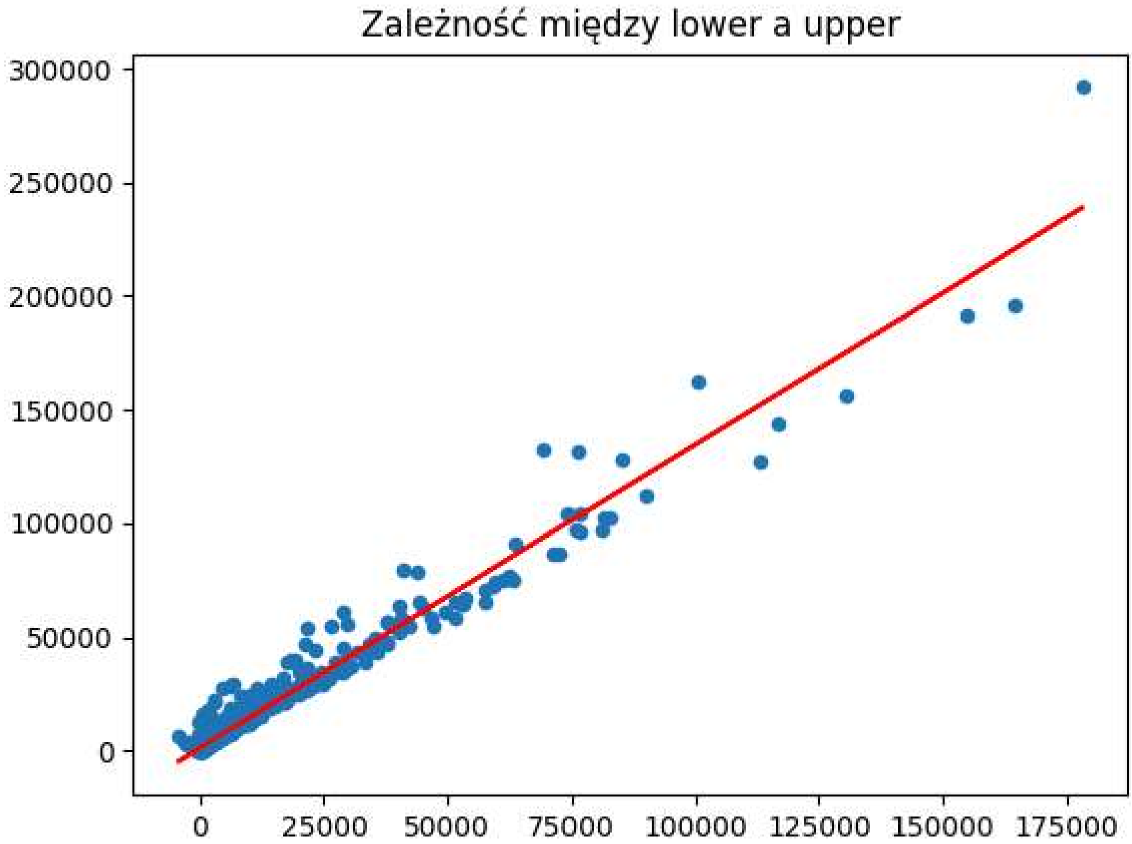
Out[8]: <Axes: xlabel='lower', ylabel='upper'>



In [10]:
```
# 5. Analiza zależności między kolumnami (z linią regresji)

import seaborn as sns
import matplotlib.pyplot as plt

numeric_df = df.select_dtypes(include=['number']) # wybierz tylko kolumny numery
sns.regplot(x='lower', y='upper', data=numeric_df, scatter_kws={'s': 20}, line_k
plt.title('Zależność między lower a upper')
plt.show()
```



In [11]:
```
# 5. Analiza zależności między kolumnami (alternatywnie)
import numpy as np
x = numeric_df['lower']
y = numeric_df['upper']
slope, intercept = np.polyfit(x, y, 1) # Oblicz współczynniki regresji (liniowa
plt.scatter(x, y, s=20)
plt.plot(x, slope * x + intercept, color='red')
plt.title('Zależność między lower a upper')
plt.show()
```



In [12]:
```
# 6. Przekształcanie danych
df['diff'] = df['upper'] - df['lower'] # Dodanie nowej kolumny

mean_diff = df.groupby('location_name')['diff'].mean() # Grupowanie wg nazwy kra
print('Średnia różnica między upper a lower dla poszczególnych krajów:')
print(mean_diff)

# Sortowanie po kolumnie year:
df = df.sort_values(by='year', ascending=True)
print(df.head())
```

```
Średnia różnica między upper a lower dla poszczególnych krajów:
location_name
United States of America    3254.912411
Name: diff, dtype: float64
      location_id            location_name  year  sex_id   sex  age_group_id  \
1208          102  United States of America  2016       3  Both           171   
1209          102  United States of America  2016       3  Both           154   
1210          102  United States of America  2016       3  Both            22   
1211          102  United States of America  2016       1  Male           158   
1212          102  United States of America  2016       1  Male           170   

     age_group_name acause                     cause_name  risk_id  \
1208       45 to 64   resp  Chronic respiratory diseases      126   
1209        65 plus   resp  Chronic respiratory diseases      126   
1210       All Ages   resp  Chronic respiratory diseases      126   
1211      <20 years     rf   Expenditure on risk factors      107   
1212       20 to 44     rf   Expenditure on risk factors      107   

                        risk_name          metric         mean        lower  \
1208            Occupational risks  2016 US Dollar  1956.870521  1630.111685   
1209            Occupational risks  2016 US Dollar  1687.262878  1288.917391   
1210            Occupational risks  2016 US Dollar  4479.253973  3846.770828   
1211  High systolic blood pressure  2016 US Dollar   326.119003   227.452936   
1212  High systolic blood pressure  2016 US Dollar  3272.311580  2611.942189   

            upper         diff  
1208  2292.750220   662.638535  
1209  2127.096398   838.179007  
1210  5176.065094  1329.294266  
1211   579.438577   351.985640  
1212  3923.028964  1311.086775  
```

In [ ]: