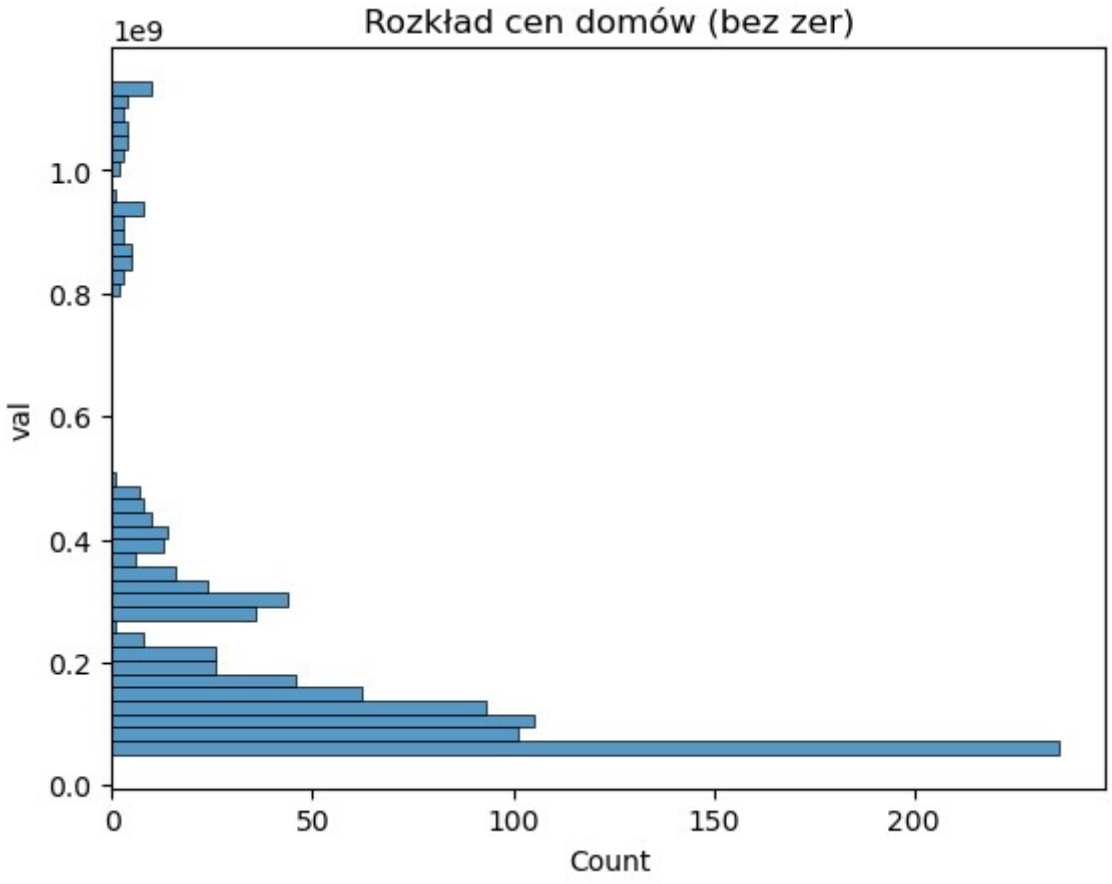
5. Wykorzystanie narzędzi do eksploracyjnej analizy danych (EDA)

In [1]:
```
import pandas as pd

# Wczytanie danych
df = pd.read_csv('IHME_GBD_2019_SMOKING_TOB_1990_2019_NUM_SMOKERS_Y2021M05D
df= df[df['val'] > 50000000]

# Podstawowe informacje o danych
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 943 entries, 0 to 14759
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   measure_name    943 non-null    object 
 1   location_id     943 non-null    int64  
 2   location_name   943 non-null    object 
 3   sex_id          943 non-null    int64  
 4   sex_name        943 non-null    object 
 5   age_group_id    943 non-null    int64  
 6   age_group_name  943 non-null    object 
 7   year_id         943 non-null    int64  
 8   val             943 non-null    float64
 9   upper           943 non-null    float64
 10  lower           943 non-null    float64
dtypes: float64(3), int64(4), object(4)
memory usage: 88.4+ KB
None
       location_id      sex_id  age_group_id      year_id           val  \
count   943.000000  943.000000         943.0   943.000000  9.430000e+02   
mean     68.375398    2.141039          29.0  2005.019088  2.070786e+08   
std      61.941784    0.957838           0.0     8.547709  2.299959e+08   
min       1.000000    1.000000          29.0  1990.000000  5.001408e+07   
25%       6.000000    1.000000          29.0  1998.000000  7.152586e+07   
50%      64.000000    3.000000          29.0  2005.000000  1.213796e+08   
75%     137.000000    3.000000          29.0  2012.000000  2.741901e+08   
max     166.000000    3.000000          29.0  2019.000000  1.144819e+09   

              upper         lower  
count  9.430000e+02  9.430000e+02  
mean   2.099499e+08  2.042090e+08  
std    2.317318e+08  2.281900e+08  
min    5.074242e+07  4.781554e+07  
25%    7.227140e+07  7.086509e+07  
50%    1.245808e+08  1.201338e+08  
75%    2.786390e+08  2.694994e+08  
max    1.157286e+09  1.131582e+09  
```

Sprawdź rozkłady danych:

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

# Filtrowanie danych, aby pominąć wartości równe 0
filtered_data = df[df['val'] > 50000000]

# Histogram rozkładu cen bez zer
sns.histplot(y=filtered_data['val'], bins=50)  # Ustawienie liczby binów
plt.title('Rozkład cen domów (bez zer)')
plt.show()
```

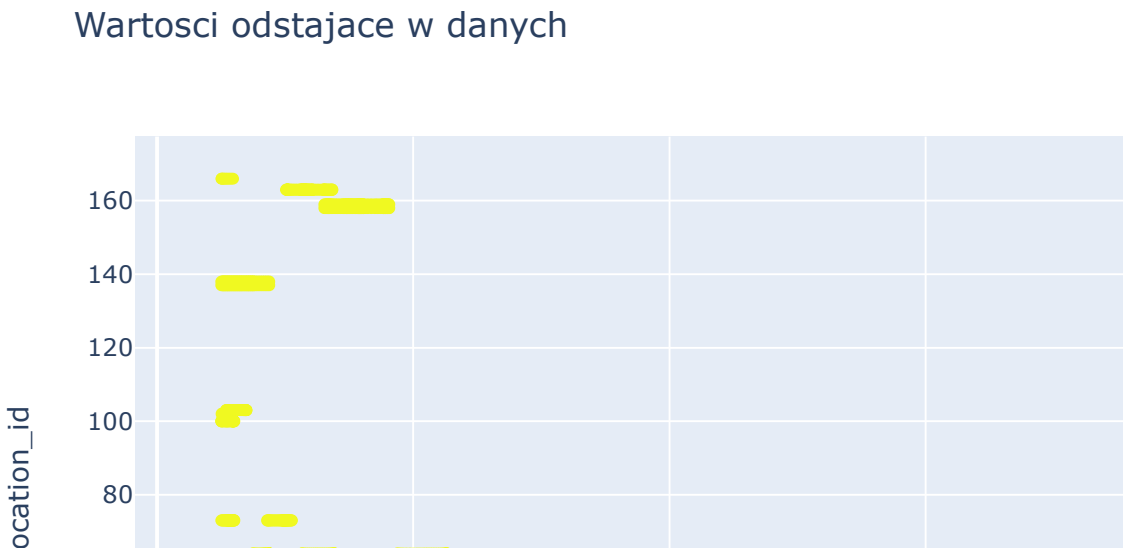


3. Detekcja wartości odstających

In [3]:
```
from sklearn.ensemble import IsolationForest

# Dopasowanie modelu Isolation Forest
isolation_forest = IsolationForest(contamination=0.1)
df['outliers'] = isolation_forest.fit_predict(df[['val']])

# Wyświetlenie wartosci odstajacych
```

```
          measure_name  location_id                              location_nam
e  \
0    Number of Smokers            1                                     Globa
l
2    Number of Smokers            1                                     Globa
l
3    Number of Smokers            1                                     Globa
l
5    Number of Smokers            1                                     Globa
l
6    Number of Smokers            1                                     Globa
l
..                 ...          ...                                       ..
.
173  Number of Smokers            4  Southeast Asia, East Asia, and Oceani
a
174  Number of Smokers            4  Southeast Asia, East Asia, and Oceani
a
176  Number of Smokers            4  Southeast Asia, East Asia, and Oceani
a
177  Number of Smokers            4  Southeast Asia, East Asia, and Oceani
a
179  Number of Smokers            4  Southeast Asia, East Asia, and Oceani
a

     sex_id sex_name  age_group_id age_group_name  year_id           val
\
0         1     Male            29      15+ years     1990  8.031015e+08
2         3     Both            29      15+ years     1990  9.922503e+08
3         1     Male            29      15+ years     1991  8.138972e+08
5         3     Both            29      15+ years     1991  1.004435e+09
6         1     Male            29      15+ years     1992  8.233148e+08
..      ...      ...           ...            ...      ...           ...
173       3     Both            29      15+ years     2017  4.828574e+08
174       1     Male            29      15+ years     2018  4.490975e+08
176       3     Both            29      15+ years     2018  4.853034e+08
177       1     Male            29      15+ years     2019  4.516942e+08
179       3     Both            29      15+ years     2019  4.881233e+08

            upper        lower  outliers
0    8.096221e+08  795908635.8        -1
2    1.000161e+09  984788043.8        -1
3    8.200339e+08  806951447.9        -1
5    1.011925e+09  996981074.1        -1
6    8.292228e+08  816726365.2        -1
..            ...          ...       ...
173  4.911908e+08  473970685.7        -1
174  4.570499e+08  440946480.5        -1
176  4.940899e+08  475985329.5        -1
177  4.602351e+08  443129996.6        -1
179  4.975039e+08  478212211.9        -1

[95 rows x 12 columns]
```

In [4]:
```
import plotly.express as px

fig = px.scatter(df, x='val', y='location_id', color='outliers', title='War
```



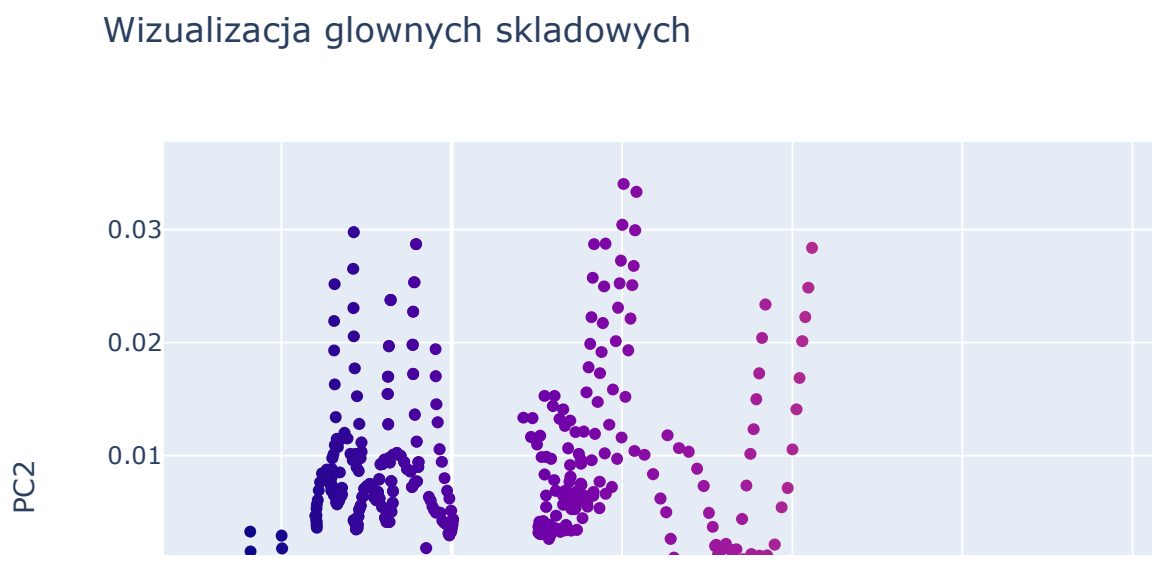
4. Analiza głównych składowych (PCA)

In [10]:
```
#PCA - Principal component analysis
#
#analiza głównych składowych, to technika statystyczna i algorytm stosowany
#PCA pozwala na uproszczenie danych wielowymiarowych, zachowując jednocześn
#w danych.

from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Skalowanie danych
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[['val', 'lower', 'upper']])

# PCA
pca = PCA(n_components=2)
principal_components = pca.fit_transform(scaled_data)

# Wynik PCA
df['PC1'] = principal_components[:, 0]
df['PC2'] = principal_components[:, 1]
```

```
[9.99970773e-01 2.91853249e-05]
```

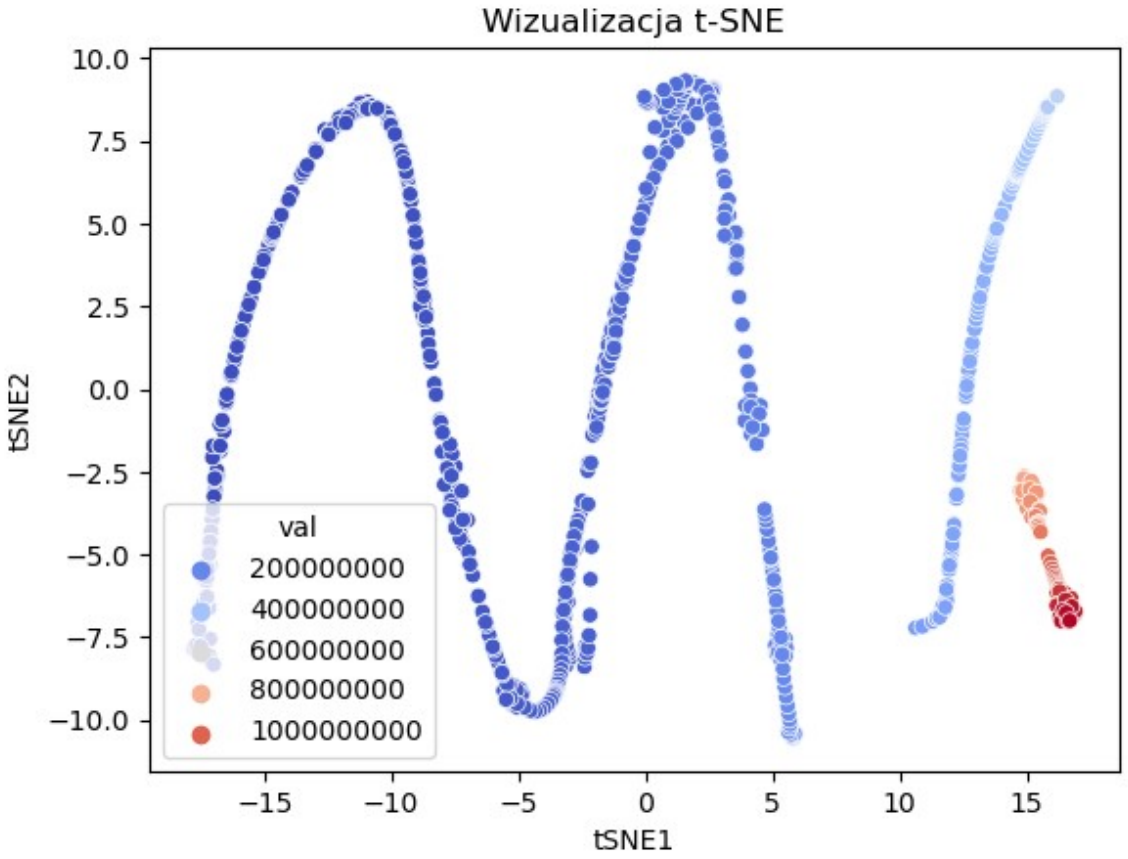
In [ ]:

In [6]:
```
fig = px.scatter(df, x='PC1', y='PC2', color='val', title='Wizualizacja glo
```



4a. Wizualizacja redukcji wymiarowości - t-SNE

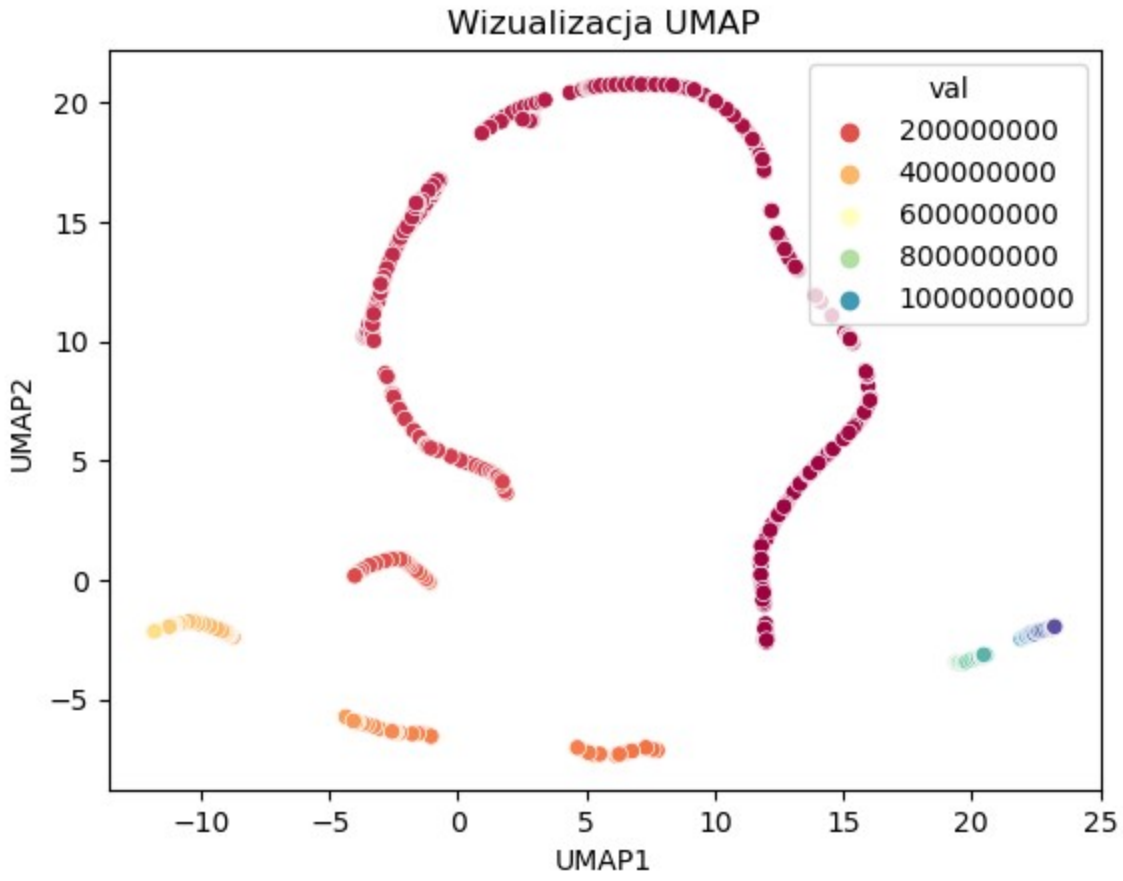
In [14]:
```
from sklearn.manifold import TSNE

# t-SNE
tsne = TSNE(n_components=3, random_state=42)
tsne_results = tsne.fit_transform(df[['val','lower','upper']])

# Dodanie wynikow do ramki danych
df['tSNE1'] = tsne_results[:, 0]
df['tSNE2'] = tsne_results[:, 1]

# Wizualizacja
sns.scatterplot(data=df, x='tSNE1', y='tSNE2', hue='val', palette='coolwarm
plt.title('Wizualizacja t-SNE')
```



4b. Wizualizacja redukcji wymiarowości - UMAP

In [20]:
```
#!pip uninstall umap
#!pip install umap-learn
import umap.umap_ as umap

# UMAP
reducer = umap.UMAP(n_neighbors=20, min_dist=0.1, random_state=42)
umap_results = reducer.fit_transform(df[['val','lower','upper']])

# Dodanie wynikow do ramki danych
df['UMAP1'] = umap_results[:, 0]
df['UMAP2'] = umap_results[:, 1]

# Wizualizacja
sns.scatterplot(data=df, x='UMAP1', y='UMAP2', hue='val', palette='Spec
plt.title('Wizualizacja UMAP')
```

```
C:\Users\Tomasz 2115\AppData\Roaming\Python\Python311\site-packages\umap\u
map_.py:1952: UserWarning:

n_jobs value 1 overridden to 1 by setting random_state. Use no seed for pa
rallelism.
```



5. Interaktywna analiza zależności

In [26]:

```
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: typing_extensions in c:\users\tomasz 2115\a
ppdata\roaming\python\python311\site-packages (4.12.2)
```

In [27]:
```
!pip install altair
```

```
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: altair in c:\users\tomasz 2115\appdata\roam
ing\python\python311\site-packages (5.5.0)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\sit
e-packages (from altair) (3.1.2)
Requirement already satisfied: jsonschema>=3.0 in c:\programdata\anaconda3
\lib\site-packages (from altair) (4.17.3)
Requirement already satisfied: narwhals>=1.14.2 in c:\users\tomasz 2115\ap
pdata\roaming\python\python311\site-packages (from altair) (1.19.1)
Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\s
ite-packages (from altair) (23.1)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\users\tomas
z 2115\appdata\roaming\python\python311\site-packages (from altair) (4.1
2.2)
Requirement already satisfied: attrs>=17.4.0 in c:\programdata\anaconda3\l
ib\site-packages (from jsonschema>=3.0->altair) (22.1.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.1
4.0 in c:\programdata\anaconda3\lib\site-packages (from jsonschema>=3.0->a
ltair) (0.18.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3
\lib\site-packages (from jinja2->altair) (2.1.1)
```

In [29]:

```
Defaulting to user installation because normal site-packages is not writea
ble

ERROR: Could not find a version that satisfies the requirement TypeIs (fro
m versions: none)
ERROR: No matching distribution found for TypeIs
```

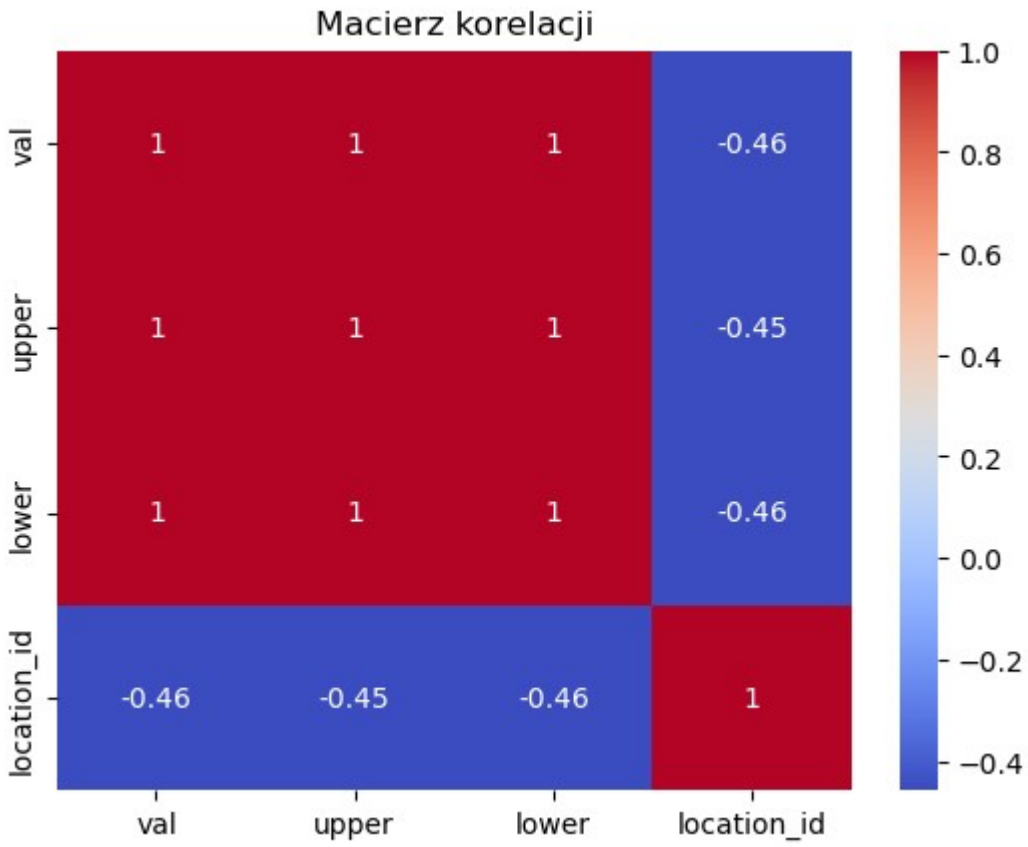
In [28]:
```
#!pip install altair
import altair as alt

chart = alt.Chart(df).mark_circle(size=60).encode(
        x='val',
        y='lower',
        color='upper',
        tooltip=['val', 'lower', 'upper']
).interactive()
```

```
---------------------------------------------------------------------------
-
ImportError                               Traceback (most recent call las
t)
Cell In[28], line 2
      1 #!pip install altair
----> 2 import altair as alt
      4 chart = alt.Chart(df).mark_circle(size=60).encode(
      5         x='val',
      6         y='lower',
      7         color='upper',
      8         tooltip=['val', 'lower', 'upper']
      9 ).interactive()
     11 chart.show()

File ~\AppData\Roaming\Python\Python311\site-packages\altair\__init__.py:6
49
    645 def __dir__():
    646     return __all__
--> 649 from altair.vegalite import *
    650 from altair.vegalite.v5.schema.core import Dict
    651 from altair.jupyter import JupyterChart

File ~\AppData\Roaming\Python\Python311\site-packages\altair\vegalite\__in
it__.py:2
      1 # ruff: noqa: F403
----> 2 from .v5 import *

File ~\AppData\Roaming\Python\Python311\site-packages\altair\vegalite\v5\_
_init__.py:2
      1 # ruff: noqa: F401, F403, F405
----> 2 from altair.expr.core import datum
      3 from altair.vegalite.v5 import api, compiler, schema
      4 from altair.vegalite.v5.api import *

File ~\AppData\Roaming\Python\Python311\site-packages\altair\expr\__init_
_.py:11
      8 import sys
      9 from typing import TYPE_CHECKING, Any
---> 11 from altair.expr.core import ConstExpression, FunctionExpression
     12 from altair.vegalite.v5.schema.core import ExprRef as _ExprRef
     14 if sys.version_info >= (3, 12):

File ~\AppData\Roaming\Python\Python311\site-packages\altair\expr\core.p
y:6
      3 import datetime as dt
      4 from typing import TYPE_CHECKING, Any, Literal, Union
----> 6 from altair.utils import SchemaBase
      8 if TYPE_CHECKING:
      9     import sys

File ~\AppData\Roaming\Python\Python311\site-packages\altair\utils\__init_
_.py:14
     12 from .deprecation import AltairDeprecationWarning, deprecated, dep
recated_warn
     13 from .html import spec_to_html
---> 14 from .plugin_registry import PluginRegistry
     15 from .schemapi import Optional, SchemaBase, SchemaLike, Undefined,
is_undefined
     17 __all__ = (
```

```
    18     "SHORTHAND_KEYS",
    19     "AltairDeprecationWarning",
  (...)
    36     "use_signature",
    37 )

File ~\AppData\Roaming\Python\Python311\site-packages\altair\utils\plugin_
registry.py:13
    11     from typing import TypeIs
    12 else:
---> 13     from typing_extensions import TypeIs
    14 if sys.version_info >= (3, 12):
    15     from typing import TypeAliasType

ImportError: cannot import name 'TypeIs' from 'typing_extensions' (C:\Prog
ramData\anaconda3\Lib\site-packages\typing_extensions.py)
```

5a. Analiza macierzy korelacji

In [31]:
```
import numpy as np

# Macierz korelacji
correlation_matrix = df[['val','upper','lower','location_id']].corr()

# Heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Macierz korelacji')
```



6. Testy statystyczne

In [32]:
```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

# Model ANOVA
model = ols('val ~ C(lower)', data=df).fit()
anova_results = anova_lm(model)
```

	df	sum_sq	mean_sq	F	PR(>F)
C(lower)	841.0	4.983002e+19	5.925092e+16	7.279439e+27	0.0
Residual	101.0	8.220885e-10	8.139490e-12	NaN	NaN

In [ ]: