

SPRAWOZDANIE

Zajęcia: Nauka o danych I

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 5 Data 22.11.2024 Temat: "Wykorzystanie narzędzi do eksploracyjnej analizy danych " Wariant 7	Tomasz Pietrzyk Informatyka II stopień, niestacjonarne, 1semestr, gr.1a
--	--

1. Polecenie: wariant 7 zadania

Proszę na podstawie własnego zbioru danych poprowadzić zaawansowaną eksploracyjną analizę danych, w tym:

- identyfikować wartości odstające za pomocą algorytmu Isolation Forest,
- redukować wymiarowość danych z użyciem PCA,
- tworzyć zaawansowane interaktywne wizualizacje danych,
- wizualizować dane wielowymiarowe za pomocą zaawansowanych algorytmów (t-SNE, UMAP),
- tworzyć interaktywne wizualizacje danych w 2D i 3D,
- analizować zależności między zmiennymi za pomocą macierzy korelacji.
- przeprowadzać testy statystyczne dla analizy różnic w grupach.

7. Global Burden of Disease Study 2019 (GBD 2019) Smoking Tobacco Use Prevalence 1990-2019 <http://ghdx.healthdata.org/record/ihme-data/gbd-2019-smoking-tobacco-use-prevalence-1990-2019>

2. Opis programu opracowanego (kody źródłowe, rzuty ekranu)

GitHub: [https://github.com/TomekPietrzyk/NOD I 2024 NS.git](https://github.com/TomekPietrzyk/NOD_I_2024_NS.git)

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_Y2021H05DZ7.CSV')
df = df[df['val'] > 50000000]

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

<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	2.141839	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	5.001400e+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

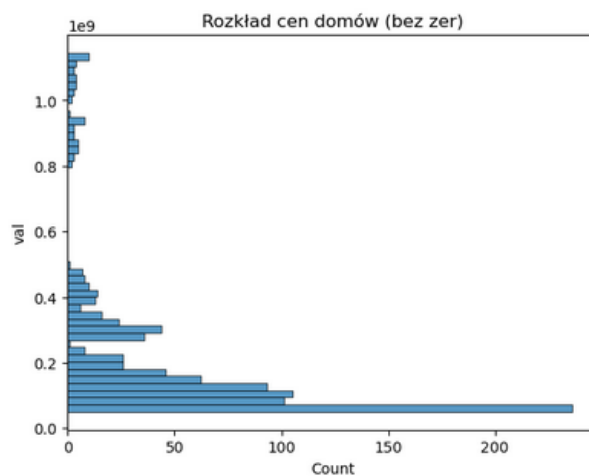
	upper	lower
count	9.430000e+02	9.430000e+02
mean	2.099499e+08	2.042090e+08
std	2.317318e+08	2.281900e+08
min	5.074242e+07	4.781554e+07
25%	7.227140e+07	7.086500e+07
50%	1.245800e+08	1.201330e+08
75%	2.786390e+08	2.694994e+08
max	1.157286e+09	1.131582e+09

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 wartości odstających
print(df[df['outliers'] == -1])
```

	measure_name	location_id	location_name	\
0	Number of Smokers	1	Global	
2	Number of Smokers	1	Global	
3	Number of Smokers	1	Global	
5	Number of Smokers	1	Global	
6	Number of Smokers	1	Global	
..	
173	Number of Smokers	4	Southeast Asia, East Asia, and Oceania	
174	Number of Smokers	4	Southeast Asia, East Asia, and Oceania	
176	Number of Smokers	4	Southeast Asia, East Asia, and Oceania	
177	Number of Smokers	4	Southeast Asia, East Asia, and Oceania	
179	Number of Smokers	4	Southeast Asia, East Asia, and Oceania	

	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	

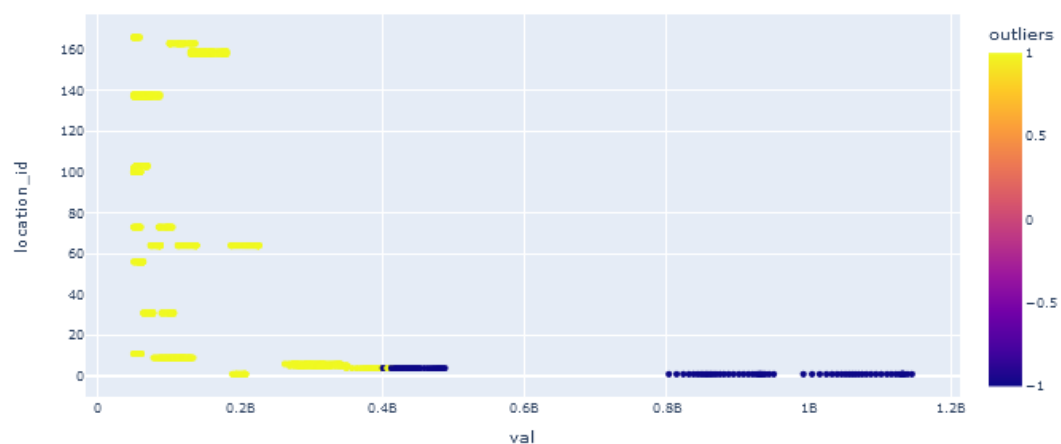
	upper	lower	outliers
0	8.096221e+08	795908635.8	-1
2	1.000161e+09	984788043.8	-1
3	8.200339e+08	806951447.9	-1
5	1.011925e+09	996981074.1	-1
6	8.292228e+08	816726365.2	-1
..
173	4.911908e+08	473970685.7	-1
174	4.570499e+08	440946480.5	-1
176	4.940899e+08	475985329.5	-1
177	4.602351e+08	443129996.6	-1
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='wartosci odstajace w danych')
fig.show()
```

Wartosci odstajace w danych



4. Analiza głównych składowych (PCA)

```
In [10]: #PCA - Principal component analysis
#
#analiza głównych składowych, to technika statystyczna i algorytm stosowany w celu redukcji wymiarowości danych.
#PCA pozwala na uproszczenie danych wielowymiarowych, zachowując jednocześnie jak najwięcej informacji o zmienności
#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]
print(pca.explained_variance_ratio_)

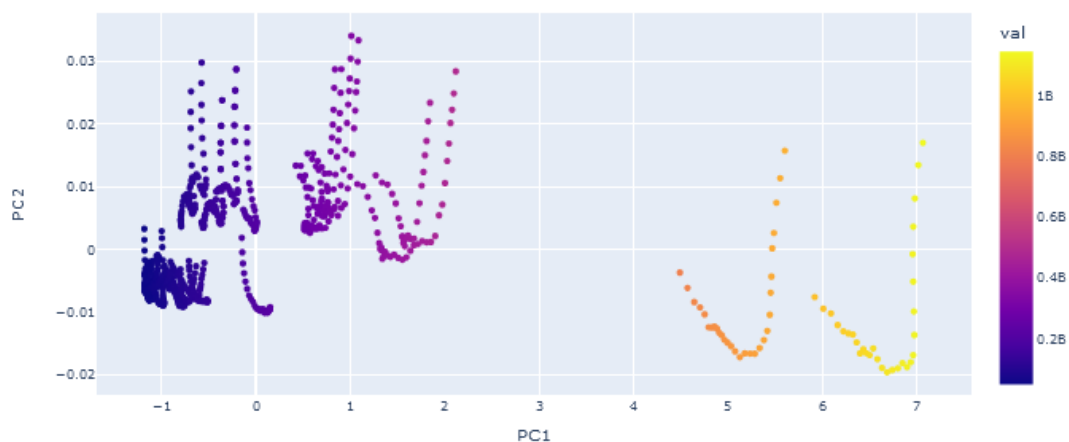
[9.99970773e-01 2.91853249e-05]
```

In []:

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



Wizualizacja głównych składowych



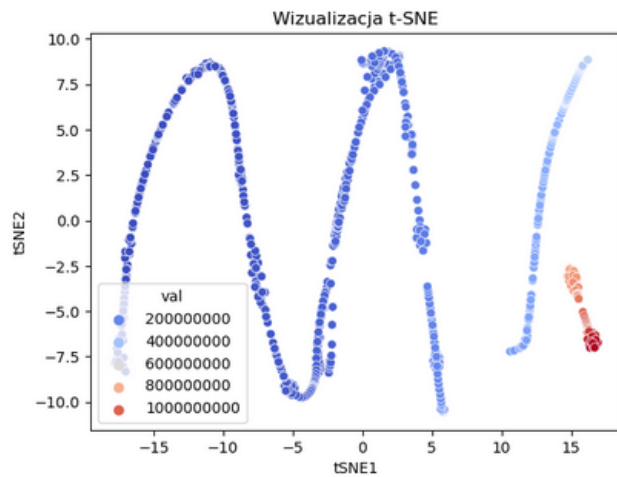
4a. Wizualizacja redukcji wymiarowości - t-SNE

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

# t-SNE
tsne = TSNE(n_components=2, random_state=42)
tsne_results = tsne.fit_transform(df[['val', 'lower', 'upper']])

# Dodanie wyników 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')
plt.show()
```



4b. Wizualizacja redukcji wymiarowości - UMAP

In [20]:

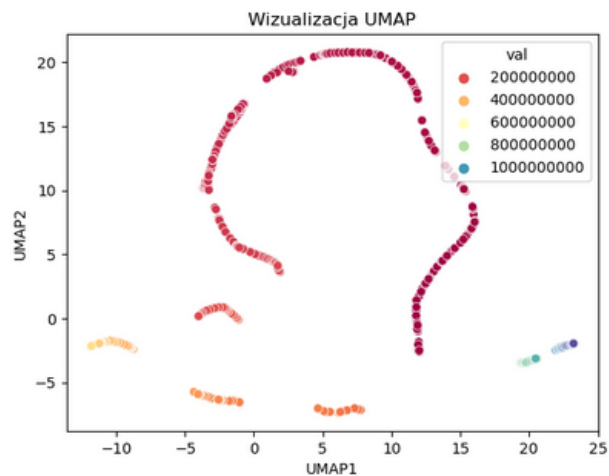
```
#!/pip uninstall umap
#!/pip install umap-learn
import umap.umap_ as umap

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

# Dodanie wyników 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='Spectral')
plt.title('Wizualizacja UMAP')
plt.show()
```

C:\Users\Tomasz.2115\AppData\Roaming\Python\Python311\site-packages\umap\umap_.py:1952: UserWarning:
n_jobs value 1 overridden to 1 by setting random_state. Use no seed for parallelism.

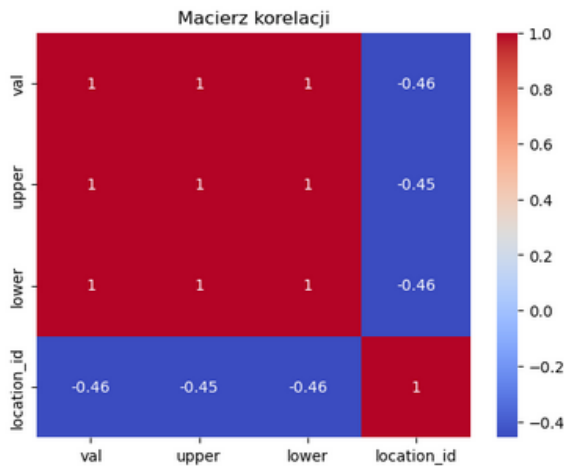


5a. Analiza macierzy korelacji

```
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')
plt.show()
```



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)

print(anova_results)
```

	df	sum_sq	mean_sq	F	PR(>F)
C(lower)	841.0	4.983882e+19	5.925882e+16	7.279439e+27	0.0
Residual	101.0	8.228885e-10	8.139490e-12	NaN	NaN

3. Wnioski

Porównanie technik:

- PCA dobrze działa w przypadku danych o strukturze liniowej, natomiast t-SNE i UMAP lepiej radzą sobie z nieliniowymi zależnościami.
- t-SNE skupia się na lokalnych relacjach, co czyni je przydatnym do analizy klastrow.
- UMAP łączy zalety t-SNE i PCA, zachowując zarówno lokalne, jak i globalne struktury danych.

Przydatność do analizy:

- Wszystkie techniki zredukowały wymiarowość danych, co ułatwiło ich eksplorację, wizualizację i przygotowanie do dalszych analiz, takich jak klasteryzacja lub klasyfikacja.

Dobór metody:

- PCA jest idealne do interpretacji zmiennych w danych.
- t-SNE i UMAP są bardziej odpowiednie do wizualizacji i analizy klastrów, przy czym UMAP oferuje większą szybkość i zachowanie globalnych relacji.