```
# Importowanie potrzebnych bibliotek
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from scipy.stats import gaussian_kde
```

1 Plik

```
# Ścieżka do pliku CSV
file_path = "C:/Users/Miłosz/Desktop/Studia/praca
inzynierska/insurance.csv"

# Wczytywanie danych z pliku CSV
try:
    data = pd.read_csv(file_path)
    print("Dane zostały pomyślnie wczytane.")
    print(data.head())
except FileNotFoundError:
    print("Plik nie został znaleziony. Sprawdź ścieżkę.")
except pd.errors.EmptyDataError:
    print("Plik jest pusty.")
except Exception as e:
    print(f"Wystąpił błąd: {e}")
```

```
Dane zostały pomyślnie wczytane.
   age     sex   bmi  children smoker     region  expenses
0   19  female  27.9         0    yes  southwest  16884.92
1   18    male  33.8         1     no  southeast   1725.55
2   28    male  33.0         3     no  southeast   4449.46
3   33    male  22.7         0     no  northwest  21984.47
4   32    male  28.9         0     no  northwest   3866.86
```

```
data.isna().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
expenses    0
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
```

```
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

```
# Zmiana nazw kolumn
nowe_nazwy = {
        'age': 'Wiek',
        'sex': 'Płeć',
        'bmi': 'BMI',
        'children': 'Dzieci',
        'smoker': 'Palacz',
        'region': 'Region',
        'expenses': 'Wydatki'
    }
data.rename(columns=nowe_nazwy, inplace=True)
print(data.head())
```

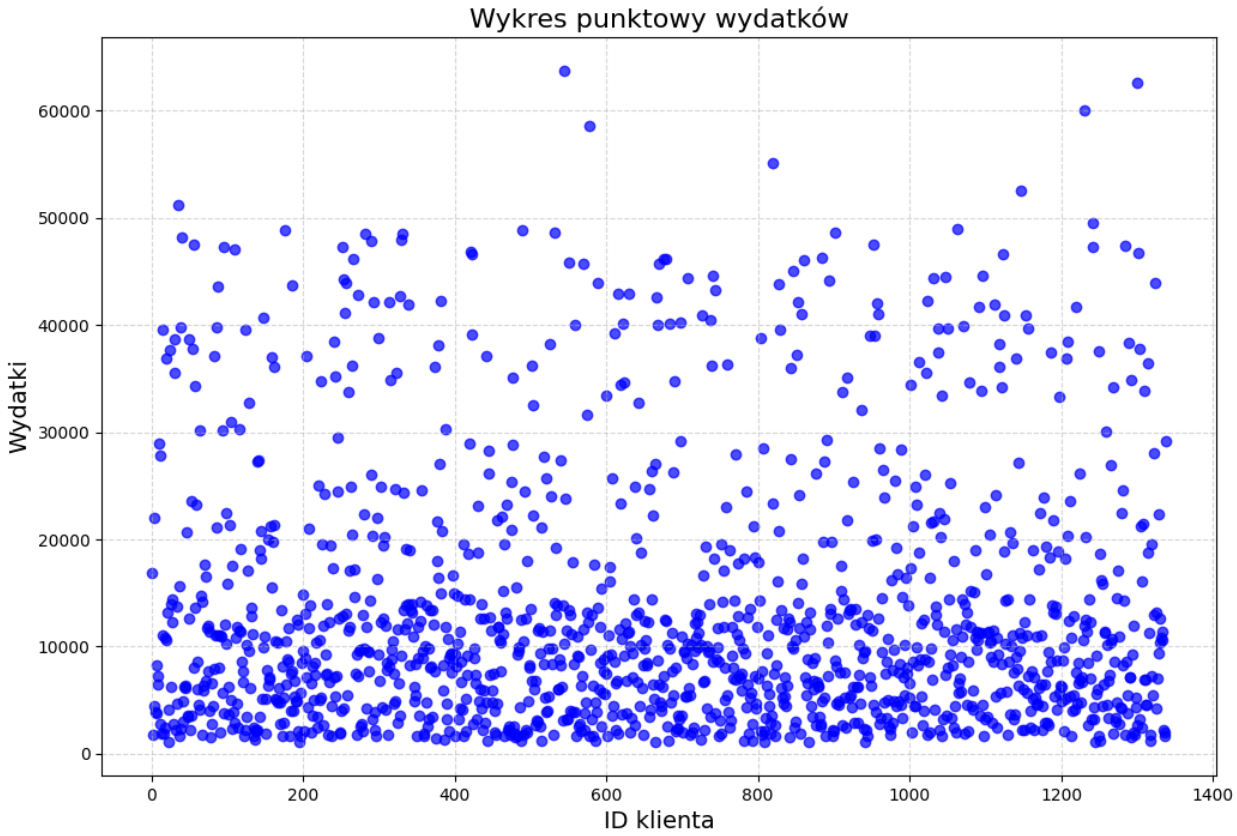
```
   Wiek    Płeć   BMI  Dzieci Palacz     Region   Wydatki
0    19  female  27.9       0    yes  southwest  16884.92
1    18    male  33.8       1     no  southeast   1725.55
2    28    male  33.0       3     no  southeast   4449.46
3    33    male  22.7       0     no  northwest  21984.47
4    32    male  28.9       0     no  northwest   3866.86
```

```
# Dodanie kolumny z ID klienta na początku tabeli
data.insert(0, 'ID', range(1, len(data) + 1))
print(data.head())
```

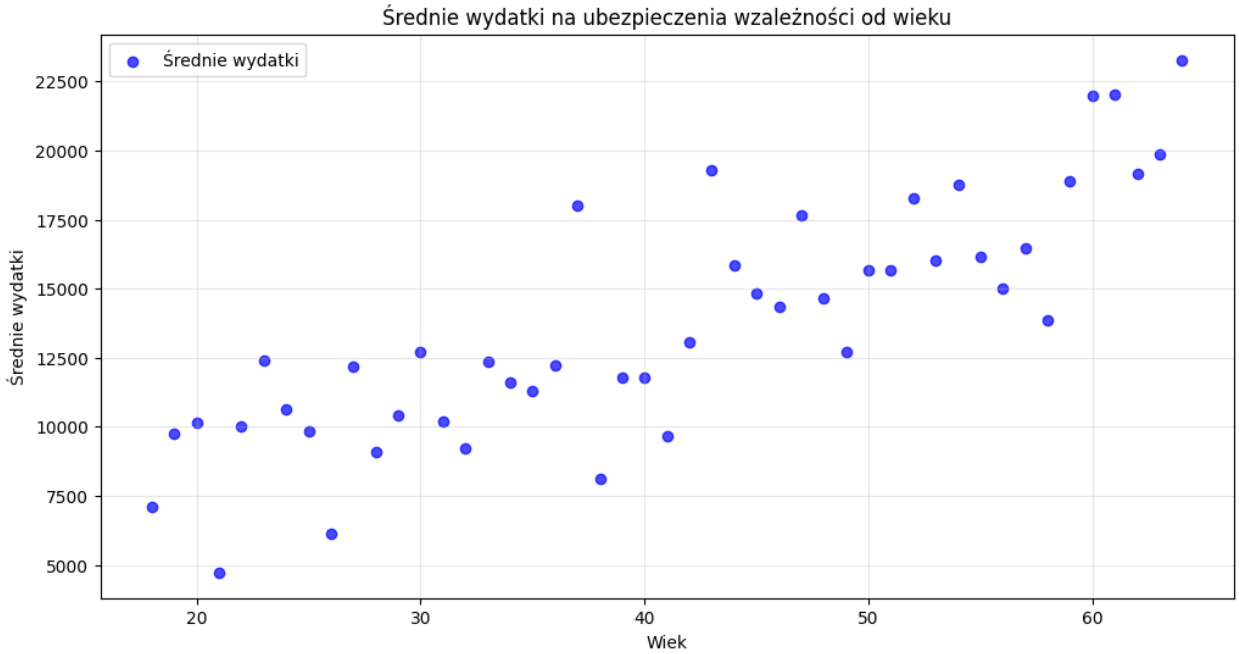
```
   ID  Wiek    Płeć   BMI  Dzieci Palacz     Region   Wydatki
0   1    19  female  27.9       0    yes  southwest  16884.92
1   2    18    male  33.8       1     no  southeast   1725.55
2   3    28    male  33.0       3     no  southeast   4449.46
3   4    33    male  22.7       0     no  northwest  21984.47
4   5    32    male  28.9       0     no  northwest   3866.86
```

```
# Tworzenie wykresu punktowego wydatków
plt.figure(figsize=(12, 8))
plt.scatter(data['ID'], data['Wydatki'], color='blue', alpha=0.7)

# Dodanie tytułu i etykiet
plt.title('Wykres punktowy wydatków', fontsize=16)
```

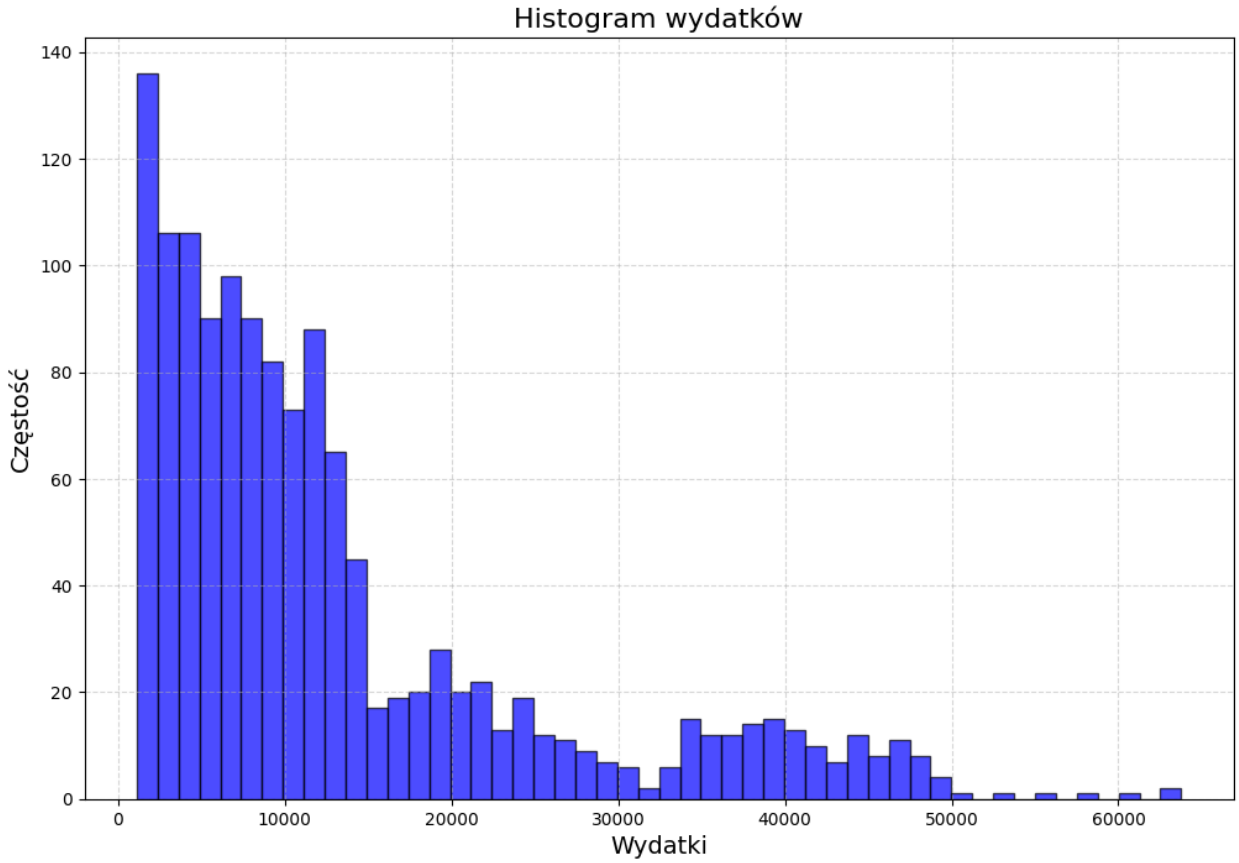
```
plt.xlabel('ID klienta', fontsize=14)
plt.ylabel('Wydatki', fontsize=14)

# Wyświetlenie wykresu
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```



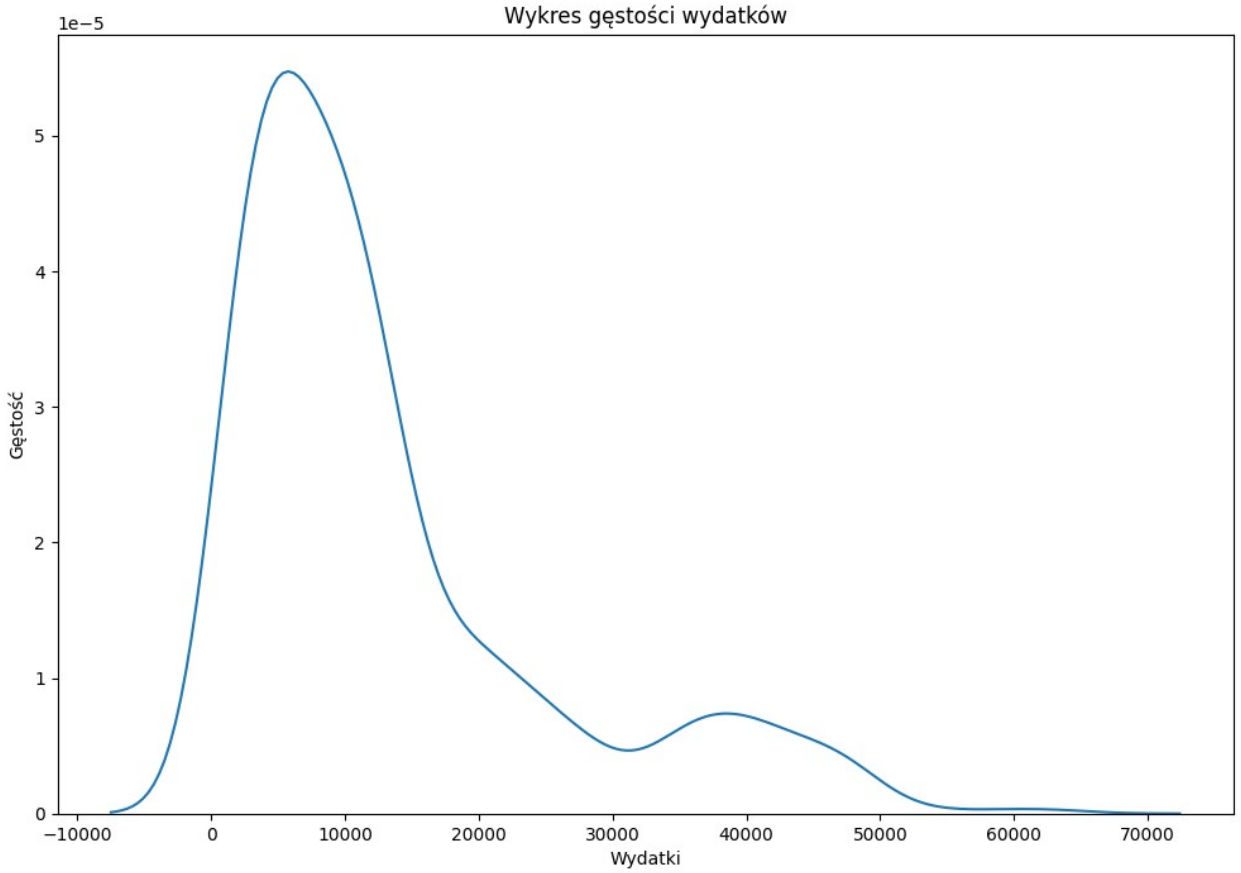

```
# Obliczenie średnich wydatków dla każdego wieku
srednie_wydatki = data.groupby('Wiek')['Wydatki'].mean()

# Rysowanie wykresu punktowego
plt.figure(figsize=(12, 6))
plt.scatter(srednie_wydatki.index, srednie_wydatki.values,
color='blue', alpha=0.7, label='Średnie wydatki')
plt.title('Średnie wydatki na ubezpieczenia wzależności od wieku')
plt.xlabel('Wiek')
plt.ylabel('Średnie wydatki')
plt.grid(alpha=0.3)
plt.legend()
plt.show()
```



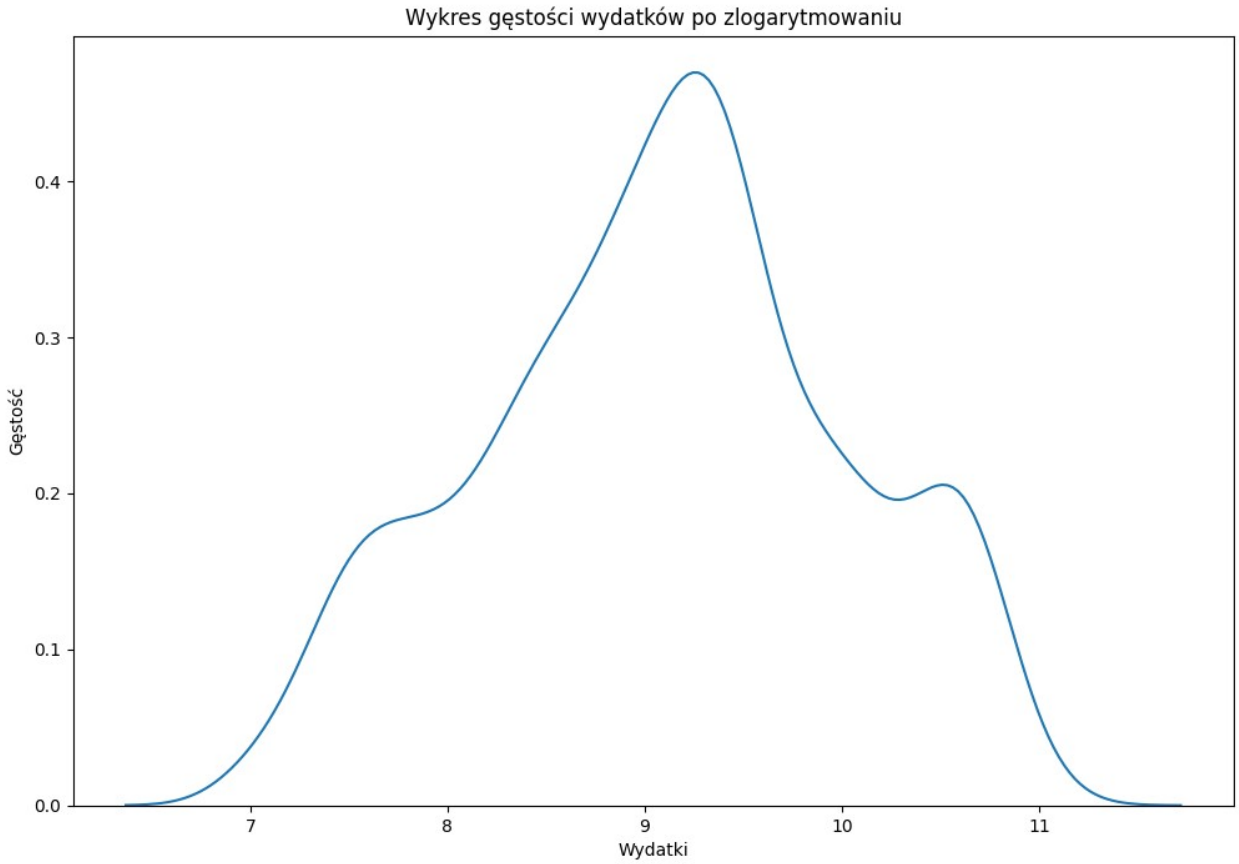

```
# Tworzenie histogramu wydatków
plt.figure(figsize=(12, 8))
plt.hist(data['Wydatki'], bins=50, color='blue', edgecolor='black',
alpha=0.7)

# Dodanie tytułu i etykiet
plt.title('Histogram wydatków', fontsize=16)
plt.xlabel('Wydatki', fontsize=14)
plt.ylabel('Częstość', fontsize=14)

# Wyświetlenie wykresu
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```




```
# Wykres gęstości wydatków
plt.figure(figsize=(12, 8))
sns.kdeplot(data['Wydatki'])
# Dodanie tytułu i opisów osi
plt.title("Wykres gęstości wydatków")
plt.xlabel("Wydatki")
plt.ylabel("Gęstość")

# Wyświetlenie wykresu
plt.show()
```



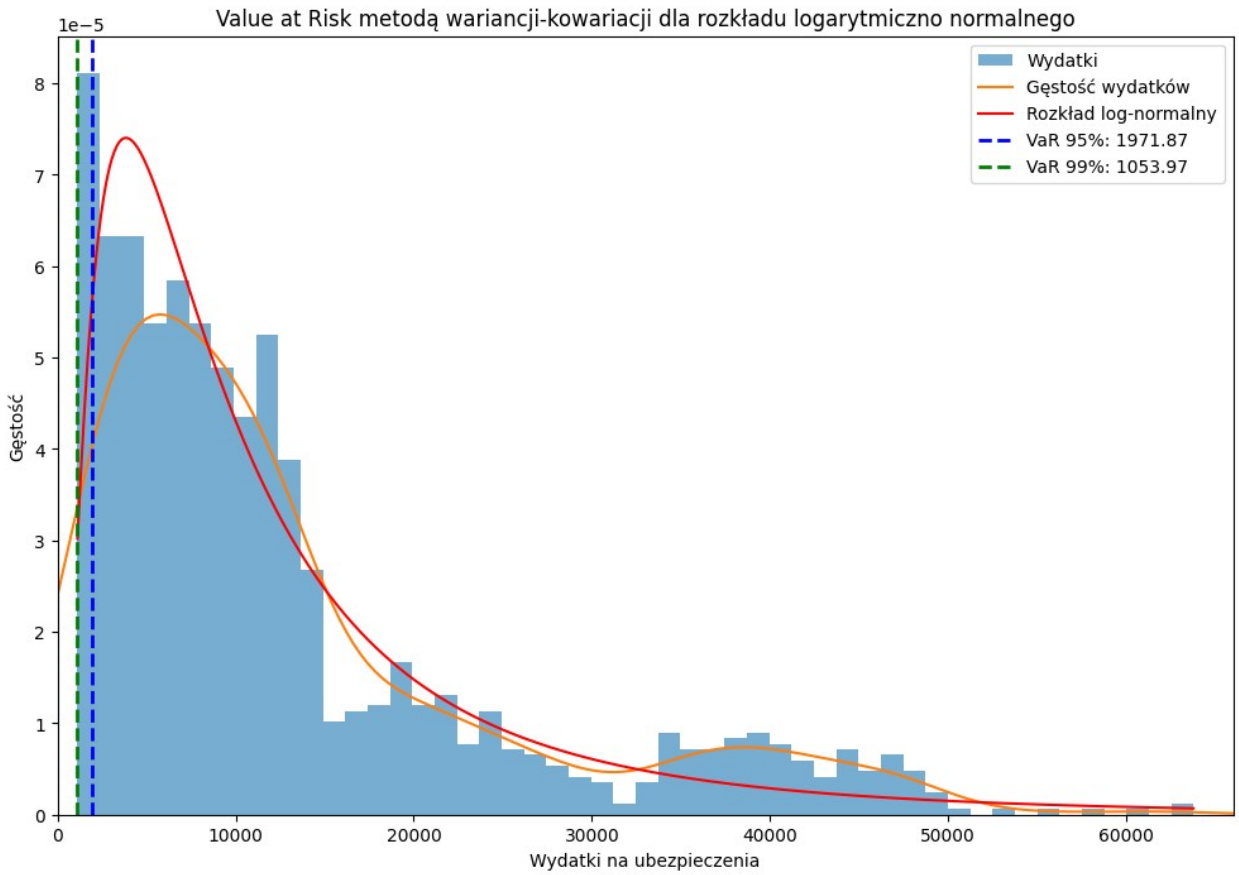

```
# Wykres gęstości wydatków
plt.figure(figsize=(12, 8))
sns.kdeplot(np.log(data['Wydatki']))
# Dodanie tytułu i opisów osi
plt.title("Wykres gęstości wydatków po zlogarytmowaniu")
plt.xlabel("Wydatki")
plt.ylabel("Gęstość")

# Wyświetlenie wykresu
plt.show()
```




```
wydatki=data["Wydatki"]

# Poziomy ufności
confidence_level_95 = 0.95
confidence_level_99 = 0.99

# Metoda wariancji-kowariancji dla rozkładu logarytmiczno-normalnego
log_data = np.log(wydatki)  # Logarytmowanie danych
mu = np.mean(log_data)
sigma = np.std(log_data)
VaR_cov_95 = np.exp(mu - stats.norm.ppf(confidence_level_95) * sigma)
VaR_cov_99 = np.exp(mu - stats.norm.ppf(confidence_level_99) * sigma)

# Wyświetlenie wyników
print(f"Value at risk metodą wariancji-kowariancji:")
print(f"  VaR 95%: {VaR_cov_95:.2f}")
print(f"  VaR 99%: {VaR_cov_99:.2f}")

# Tworzenie wykresu
plt.figure(figsize=(12, 8))
plt.hist(wydatki, bins=50, alpha=0.6, label="Wydatki", density=True)
x = np.linspace(min(wydatki), max(wydatki), 1000)
pdf = stats.lognorm.pdf(x, s=sigma, scale=np.exp(mu))
```

```
sns.kdeplot(data['Wydatki'], label="Gęstość wydatków")
plt.plot(x, pdf, 'r-', label="Rozkład log-normalny")
plt.axvline(VaR_cov_95, color='blue', linestyle='dashed', linewidth=2,
label=f"VaR 95%: {VaR_cov_95:.2f}")
plt.axvline(VaR_cov_99, color='green', linestyle='dashed',
linewidth=2, label=f"VaR 99%: {VaR_cov_99:.2f}")
plt.xlim(0, 66000)
plt.title("Value at Risk metodą wariancji-kowariacji dla rozkładu
logarytmiczno normalnego")
plt.xlabel("Wydatki na ubezpieczenia")
plt.ylabel("Gęstość")
plt.legend()
plt.show()
```

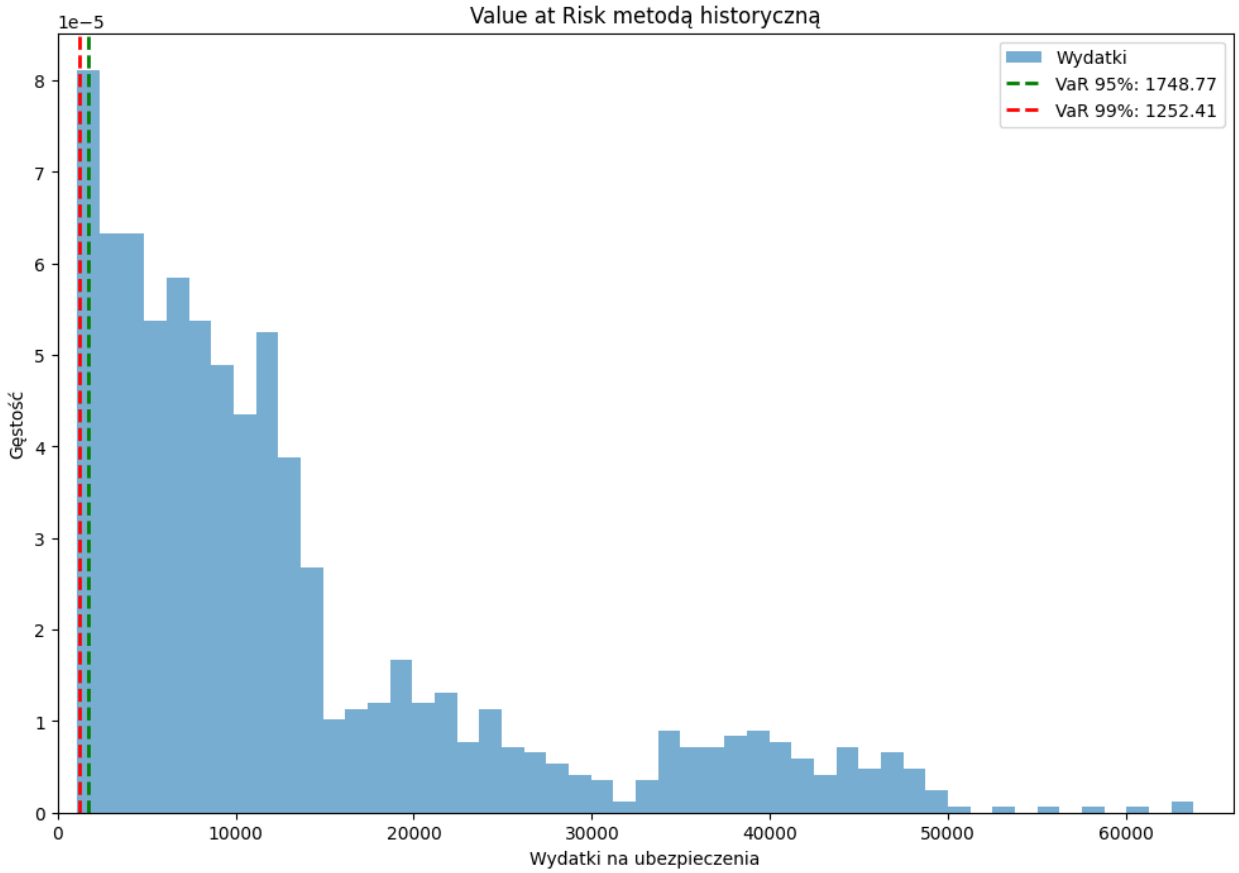
```
Value at risk metodą wariancji-kowariancji:
  VaR 95%: 1971.87
  VaR 99%: 1053.97
```




```
# Poziomy ufności
confidence_level_95 = 0.95
confidence_level_99 = 0.99
```

```python
# Metoda historyczna
sorted_data = np.sort(wydatki)
VaR_hist_95 = sorted_data[int((1 - confidence_level_95) *
len(sorted_data))]
VaR_hist_99 = sorted_data[int((1 - confidence_level_99) *
len(sorted_data))]

# Wyświetlanie wyników
print(f"Value at risk metodą historyczną:")
print(f"  VaR 95%: {VaR_hist_95:.2f}")
print(f"  VaR 99%: {VaR_hist_99:.2f}")

# Tworzenie wykresu
plt.figure(figsize=(12, 8))
plt.hist(wydatki, bins=50, alpha=0.6, label="Wydatki", density=True)
plt.axvline(VaR_hist_95, color='green', linestyle='dashed',
linewidth=2, label=f"VaR 95%: {VaR_hist_95:.2f}")
plt.axvline(VaR_hist_99, color='red', linestyle='dashed', linewidth=2,
label=f"VaR 99%: {VaR_hist_99:.2f}")
plt.xlim(0, 66000)
plt.title("Value at Risk metodą historyczną")
plt.xlabel("Wydatki na ubezpieczenia")
plt.ylabel("Gęstość")
plt.legend()
plt.show()
```

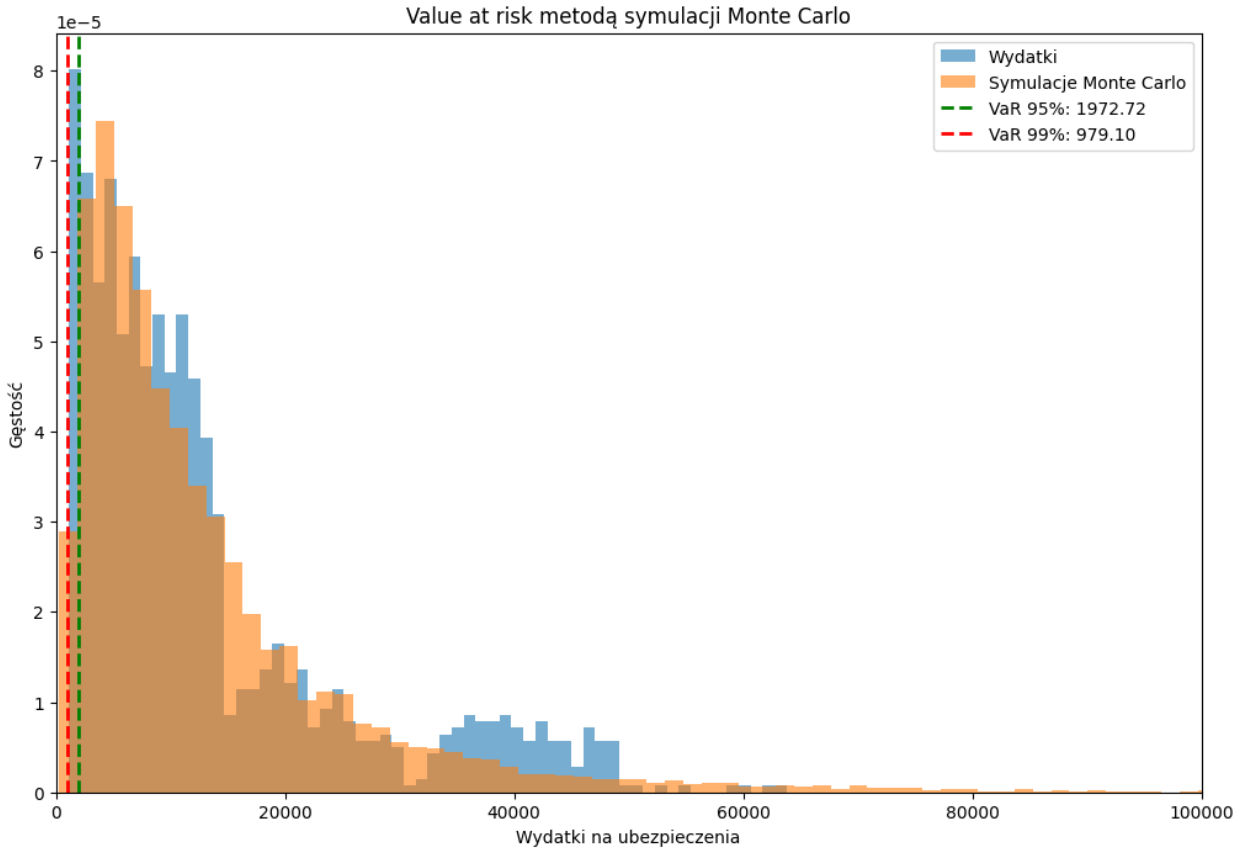
```
Value at risk metodą historyczną:
  VaR 95%: 1748.77
  VaR 99%: 1252.41
```




```
# Poziomy ufności
confidence_level_95 = 0.95
confidence_level_99 = 0.99

# Symulacja Monte Carlo
n_simulations = 10000
simulated_data = np.random.lognormal(mu, sigma, n_simulations)
VaR_mc_95 = np.percentile(simulated_data, (1 - confidence_level_95) *
100)
VaR_mc_99 = np.percentile(simulated_data, (1 - confidence_level_99) *
100)

# Wyświetlenie wyników
print(f"Value at risk metodą symulacji Monte Carlo:")
print(f"  VaR 95%: {VaR_mc_95:.2f}")
print(f"  VaR 99%: {VaR_mc_99:.2f}")

# Wizualizacja danych historycznych i Monte Carlo
plt.figure(figsize=(12, 8))
plt.hist(wydatki, bins=60, alpha=0.6, label="Wydatki", density=True)
plt.hist(simulated_data, bins=240, alpha=0.6, label="Symulacje Monte
Carlo", density=True)
plt.axvline(VaR_mc_95, color='green', linestyle='dashed', linewidth=2,
```

```
label=f"VaR 95%: {VaR_mc_95:.2f}")
plt.axvline(VaR_mc_99, color='red', linestyle='dashed', linewidth=2,
label=f"VaR 99%: {VaR_mc_99:.2f}")
plt.xlim(0, 100000)
plt.title("Value at risk metodą symulacji Monte Carlo")
plt.xlabel("Wydatki na ubezpieczenia")
plt.ylabel("Gęstość")
plt.legend()
plt.show()
```

```
Value at risk metodą symulacji Monte Carlo:
  VaR 95%: 1972.72
  VaR 99%: 979.10
```



2 Plik

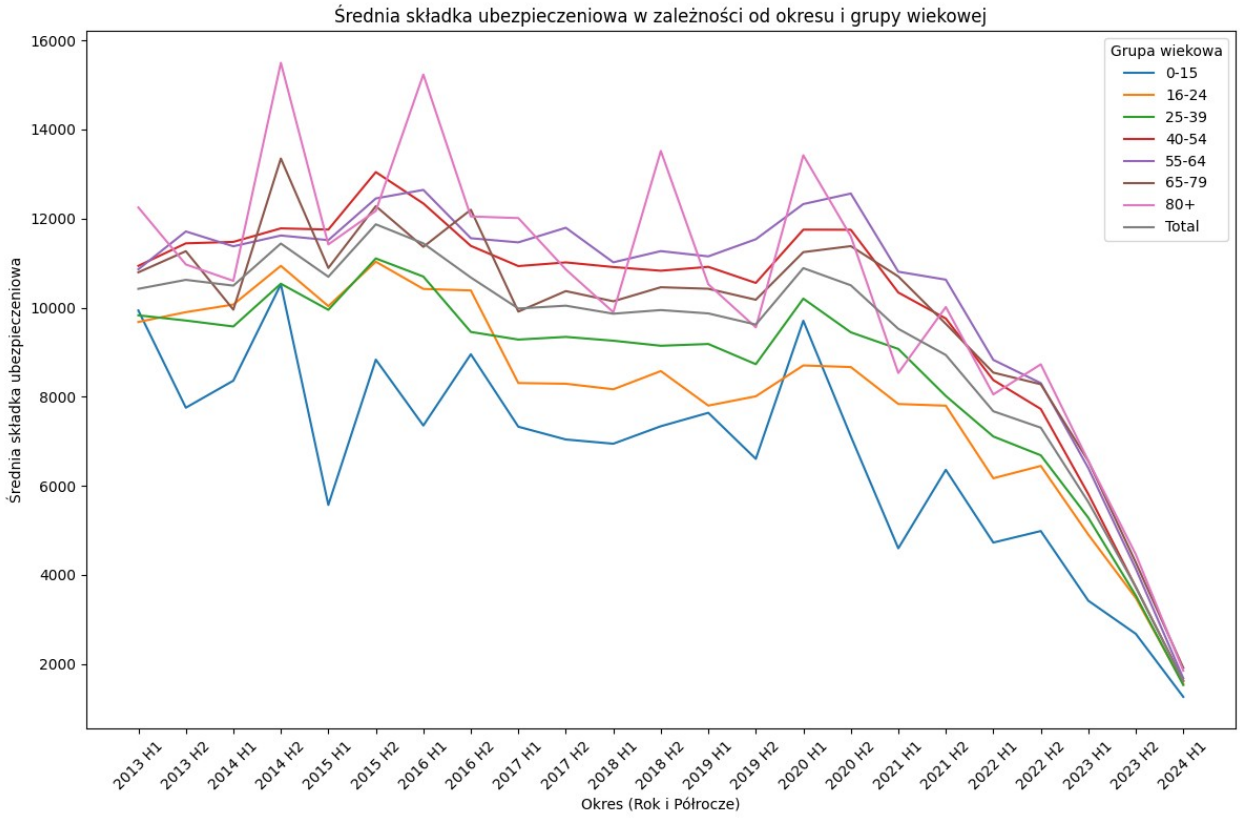
```
# Ścieżka do pliku CSV
file_path = "C:/Users/Miłosz/Desktop/Studia/praca
inzynierska/canada.csv"

# Wczytywanie danych z pliku CSV
try:
```

```
    dane = pd.read_csv(file_path)
    print("Dane zostały pomyślnie wczytane.")
    print(dane.head())
except FileNotFoundError:
    print("Plik nie został znaleziony. Sprawdź ścieżkę.")
except pd.errors.EmptyDataError:
    print("Plik jest pusty.")
except Exception as e:
    print(f"Wystąpił błąd: {e}")
```

```
Dane zostały pomyślnie wczytane.
    Age  Half_Year  Claimant_Count  Share_on_Claimant_Count
Insurer_Paid  \
0  0-15      20131             977                     0.04
9708413.47
1  0-15      20132            1242                     0.04
9629730.22
2  0-15      20141             889                     0.03
7432033.50
3  0-15      20142            1338                     0.04
14081337.89
4  0-15      20151             979                     0.03
5453924.57

   Share_on_Insurer_Paid  Average_Insurer_Paid_per_Claimant
0                   0.03                            9936.96
1                   0.03                            7753.41
2                   0.02                            8359.99
3                   0.04                           10524.17
4                   0.02                            5570.91
```

```
dane['Rok'] = dane['Half_Year'].apply(lambda x: str(x)[:4])
dane['Półrocze'] = dane['Half_Year'].apply(lambda x: 1 if str(x)[4] ==
'1' else 2)

# Stworzenie osi czasu - połączenie roku i półrocza
dane['Czas'] = dane['Rok'] + ' H' + dane['Półrocze'].astype(str)

# Wykres
plt.figure(figsize=(12, 8))

# Iterowanie przez różne grupy wiekowe
for grupa in dane['Age'].unique():
    grupa_df = dane[dane['Age'] == grupa]
    plt.plot(grupa_df['Czas'],
grupa_df['Average_Insurer_Paid_per_Claimant'], label=f'{grupa}')

plt.title('Średnia składka ubezpieczeniowa w zależności od okresu i
grupy wiekowej')
plt.xlabel('Okres (Rok i Półrocze)')
```

```
plt.ylabel('Średnia składka ubezpieczeniowa')
plt.xticks(rotation=45)
plt.legend(title='Grupa wiekowa')
plt.tight_layout()
plt.show()
```

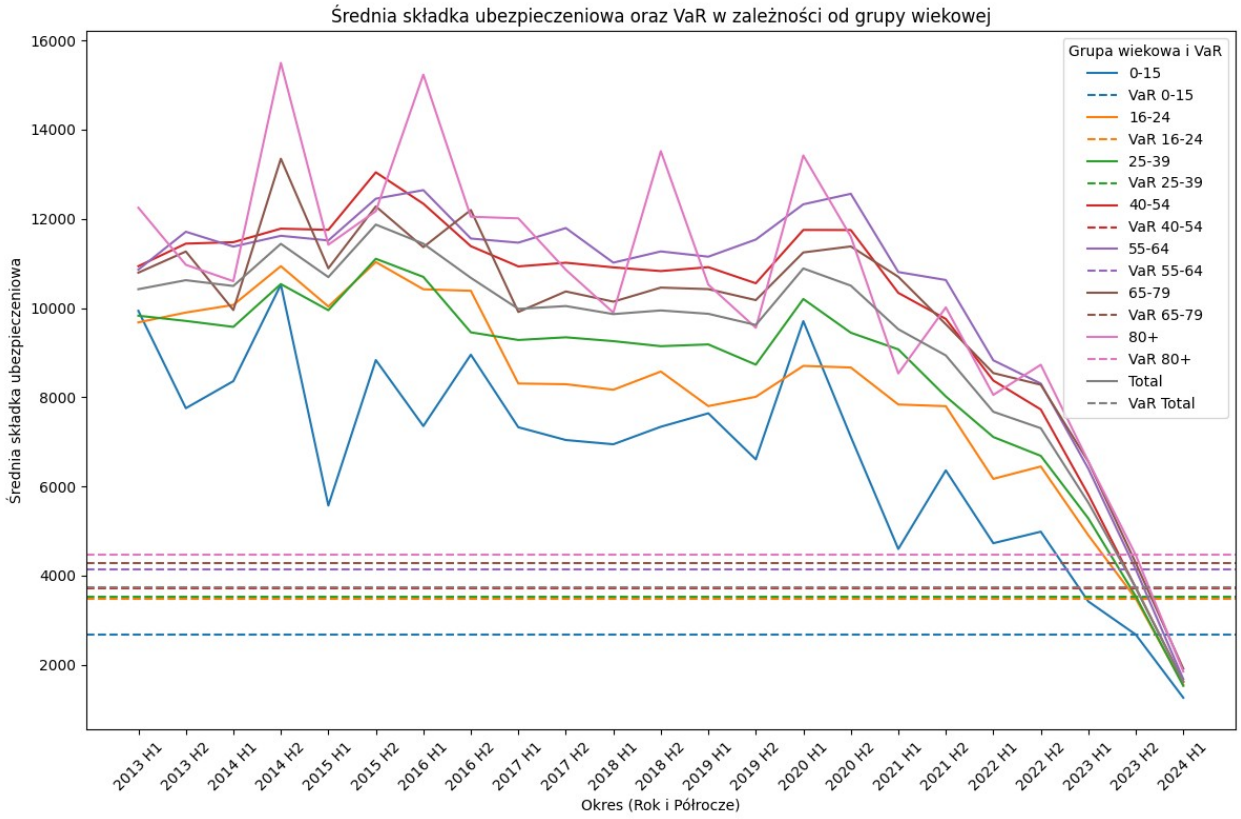



```
# Funkcja obliczająca VaR dla danej grupy wiekowej
def calculate_var(group_data, confidence_level=0.95):
    # Obliczamy VaR na podstawie metodologii historycznej (Percentyl)
    # Posortowanie składek
    sorted_data = np.sort(group_data)

    # Obliczenie wartości percentyla (np. 5% dla 95% poziomu ufności)
    percentile = (1 - confidence_level) * len(sorted_data)
    var_value = sorted_data[int(percentile)]

    return var_value

# Obliczanie VaR dla każdej grupy wiekowej
var_values = {}

for group in dane['Age'].unique():
    group_df = dane[dane['Age'] == group]
    var_value =
```

```
calculate_var(group_df['Average_Insurer_Paid_per_Claimant'],
confidence_level=0.95)
    var_values[group] = var_value

# Wyświetlanie wyników
for group, var in var_values.items():
    print(f"Value at Risk (VaR) metodą historyczną dla grupy wiekowej
{group}: {var}")
```

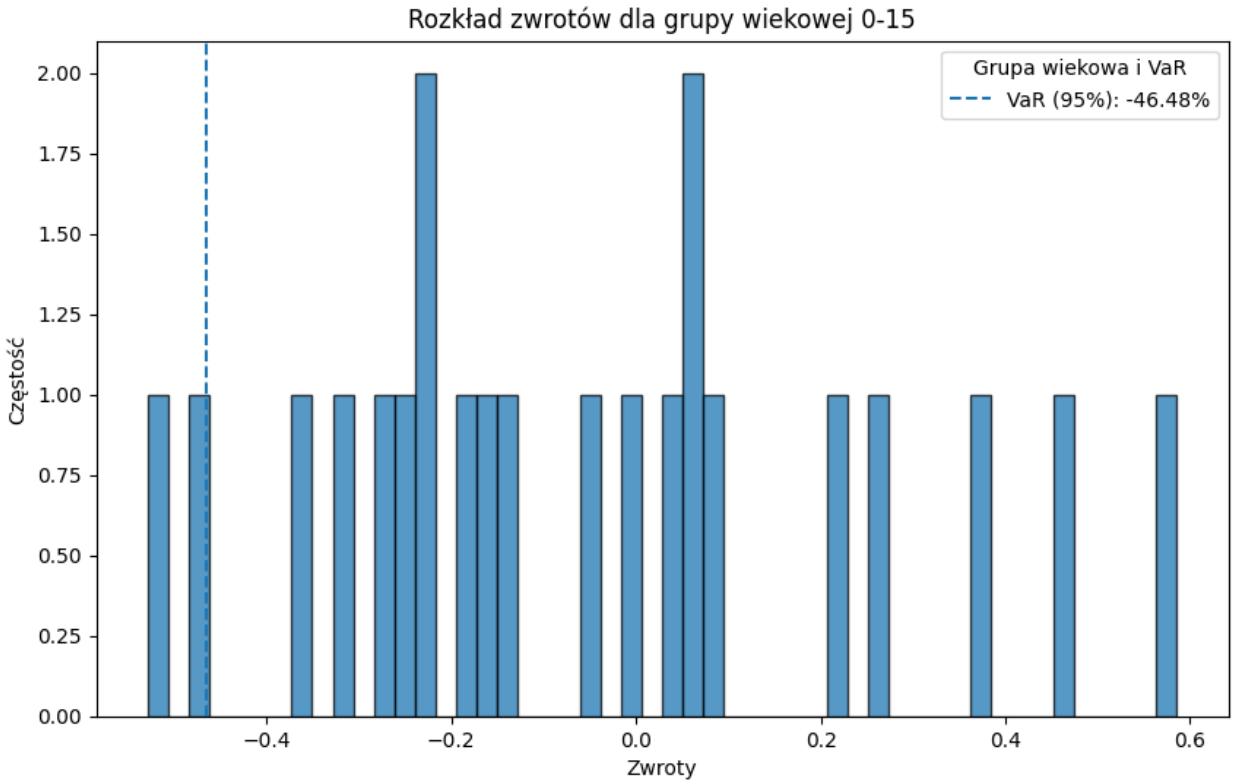
```
Value at Risk (VaR) dla grupy wiekowej 0-15: 2675.88
Value at Risk (VaR) dla grupy wiekowej 16-24: 3480.87
Value at Risk (VaR) dla grupy wiekowej 25-39: 3535.1
Value at Risk (VaR) dla grupy wiekowej 40-54: 3725.08
Value at Risk (VaR) dla grupy wiekowej 55-64: 4136.59
Value at Risk (VaR) dla grupy wiekowej 65-79: 4277.38
Value at Risk (VaR) dla grupy wiekowej 80+: 4458.25
Value at Risk (VaR) dla grupy wiekowej Total: 3726.99
```

```
# Wykres
plt.figure(figsize=(12, 8))

# Iterowanie przez różne grupy wiekowe
for grupa in dane['Age'].unique():
    grupa_df = dane[dane['Age'] == grupa]

    # Kolor dla danej grupy
    color = plt.cm.tab10(list(dane['Age'].unique()).index(grupa))

    # Rysowanie linii z danymi dla grupy wiekowej
    plt.plot(grupa_df['Czas'],
grupa_df['Average_Insurer_Paid_per_Claimant'], label=f'{grupa}',
color=color)

    # Dodanie poziomej linii VaR dla danej grupy wiekowej w tym samym
kolorze
    plt.axhline(y=var_values[grupa], color=color, linestyle='--',
label=f'VaR {grupa}')

plt.title('Średnia składka ubezpieczeniowa oraz VaR w zależności od
grupy wiekowej')
plt.xlabel('Okres (Rok i Półrocze)')
plt.ylabel('Średnia składka ubezpieczeniowa')
plt.xticks(rotation=45)
plt.legend(title='Grupa wiekowa i VaR', loc='upper right')
plt.tight_layout()
plt.show()
```

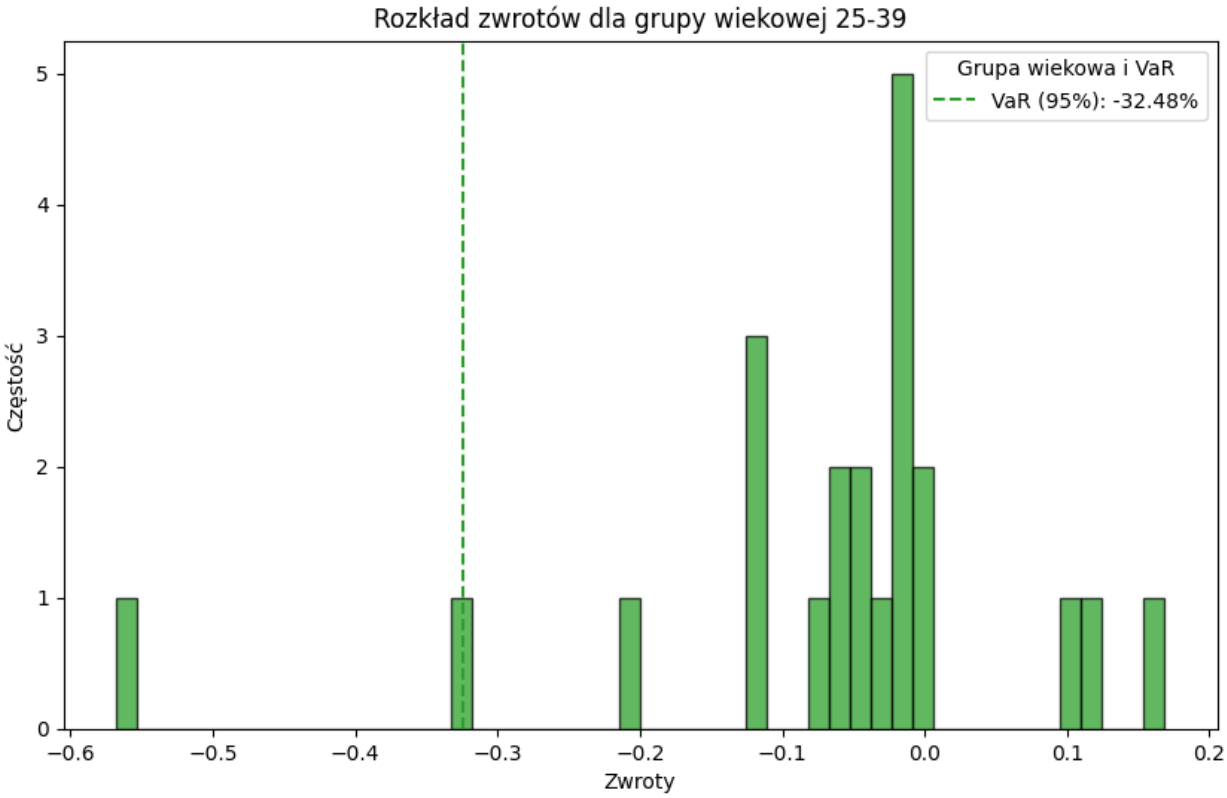


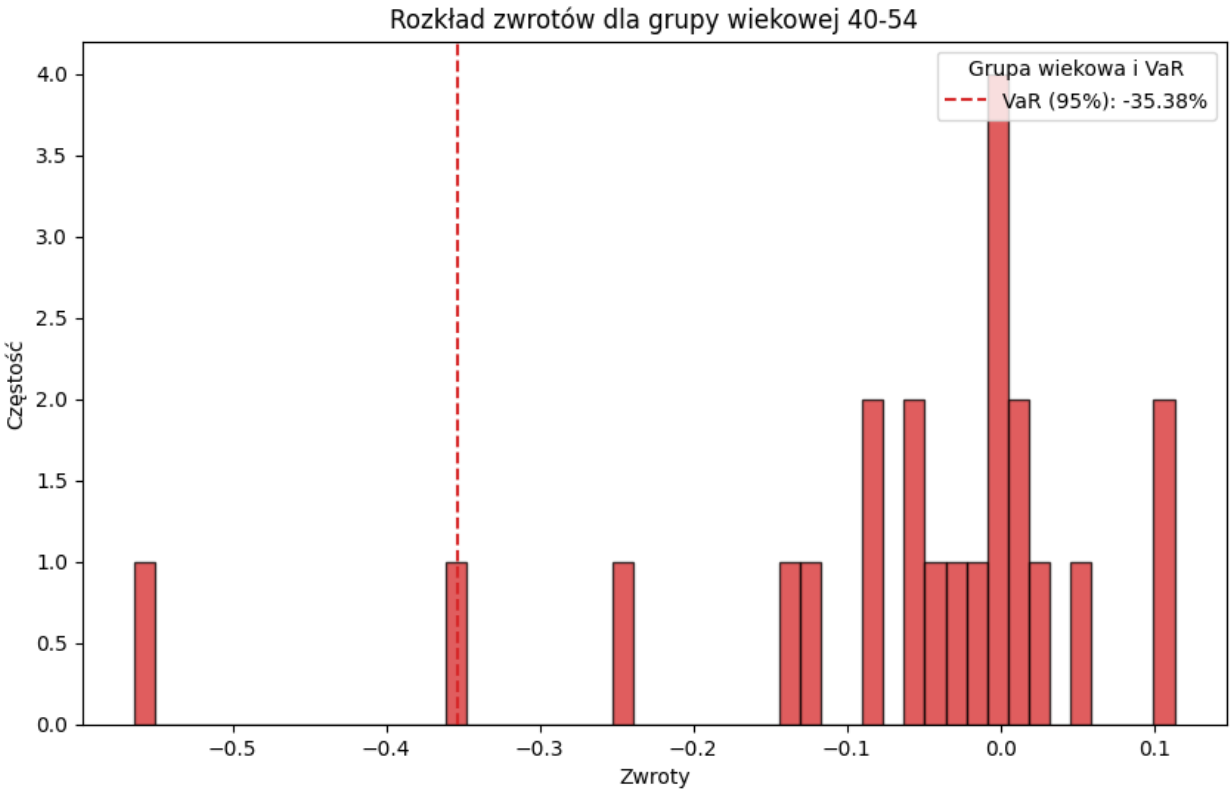

```
# Wykres
# Iterowanie przez różne grupy wiekowe
for grupa in dane['Age'].unique():
    grupa_df = dane[dane['Age'] == grupa]
    color = plt.cm.tab10(list(dane['Age'].unique()).index(grupa))
    returns =
grupa_df['Average_Insurer_Paid_per_Claimant'].pct_change().dropna()
    confidence_level = 0.95
    VaR_historical = np.percentile(returns, (1 - confidence_level) *
100)
    print(f"VaR metodą historycznaą dla grupy wiekowej {grupa}:
{VaR_historical:.2%}")
    plt.figure(figsize=(10, 6))
    plt.title('Rozkład zwrotów dla grupy wiekowej ' f'{grupa}')
    plt.xlabel('Zwroty')
    plt.ylabel('Częstość')
    plt.hist(returns, bins=50, alpha=0.75, color=color,
edgecolor='black')
    plt.axvline(VaR_historical, linestyle='--', color=color,
label=f'VaR (95%): {VaR_historical:.2%}')
    plt.legend(title='Grupa wiekowa i VaR', loc='upper right')

VaR metodą historycznaą dla grupy wiekowej 0-15: -46.48%
VaR metodą historycznaą dla grupy wiekowej 16-24: -28.78%
VaR metodą historycznaą dla grupy wiekowej 25-39: -32.48%
```

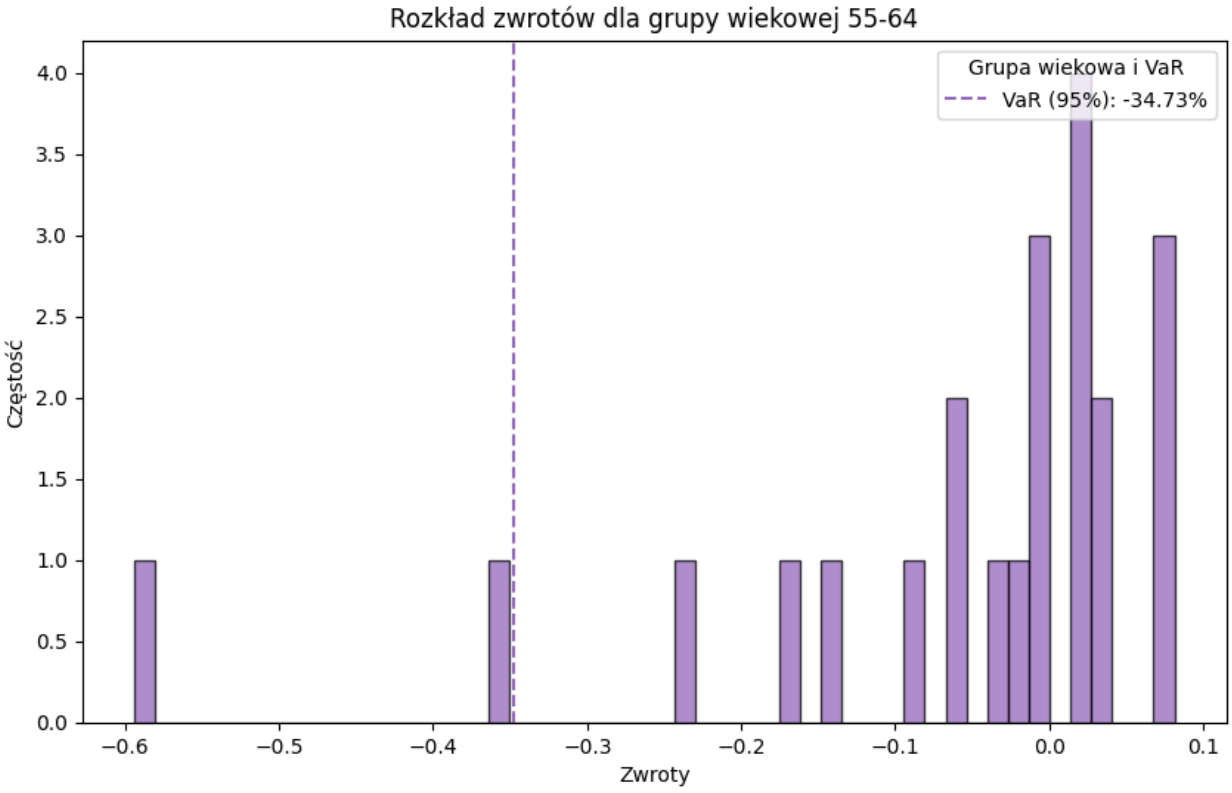
```
VaR metodą historycznaą dla grupy wiekowej 40-54: -35.38%
VaR metodą historycznaą dla grupy wiekowej 55-64: -34.73%
VaR metodą historycznaą dla grupy wiekowej 65-79: -34.05%
VaR metodą historycznaą dla grupy wiekowej 80+: -31.77%
VaR metodą historycznaą dla grupy wiekowej Total: -33.32%
```

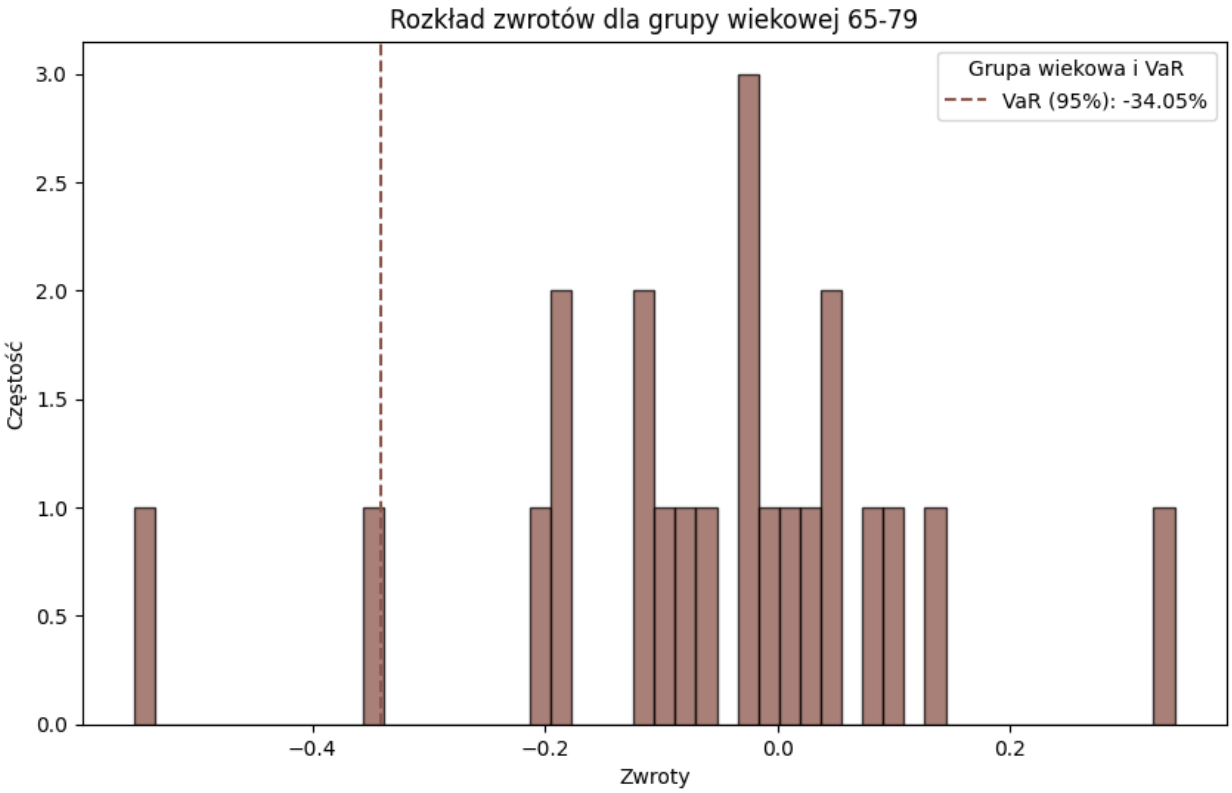


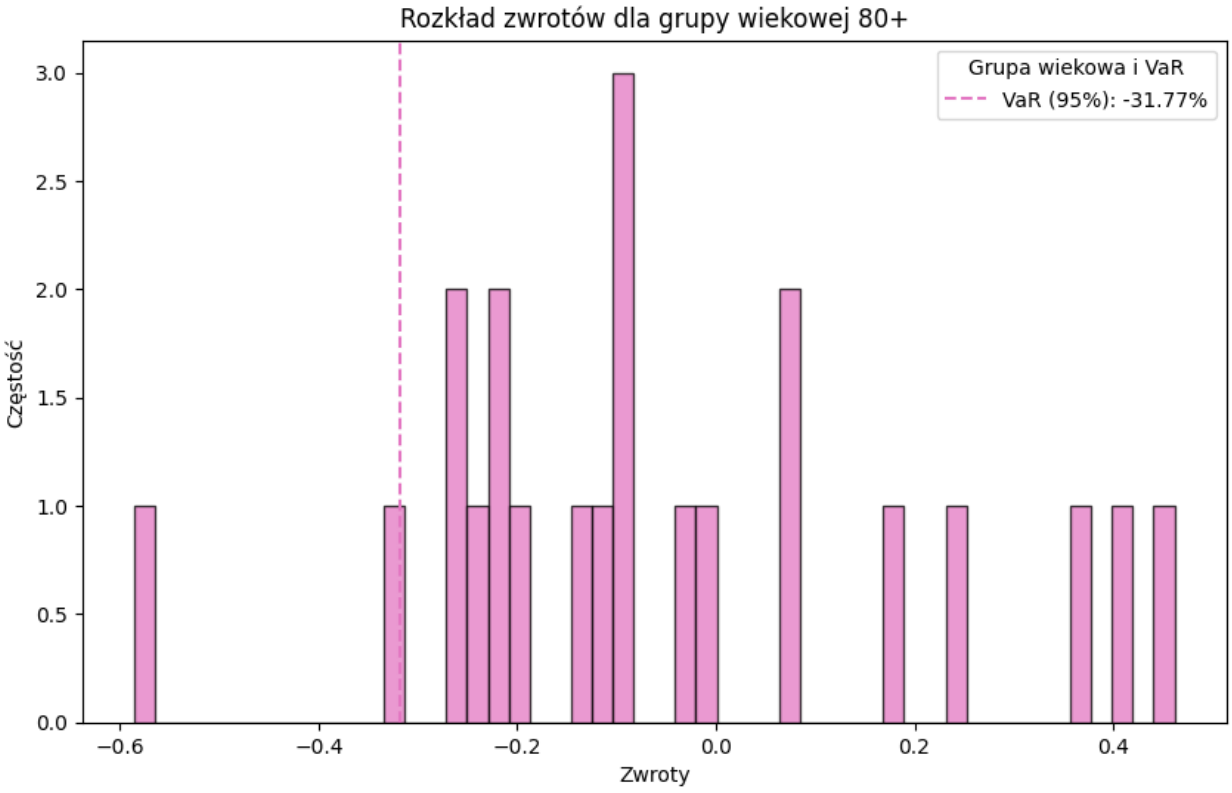

Rozkład zwrotów dla grupy wiekowej 16-24
Grupa wiekowa i VaR
VaR (95%): -28.78%
Częstość
Zwroty
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1

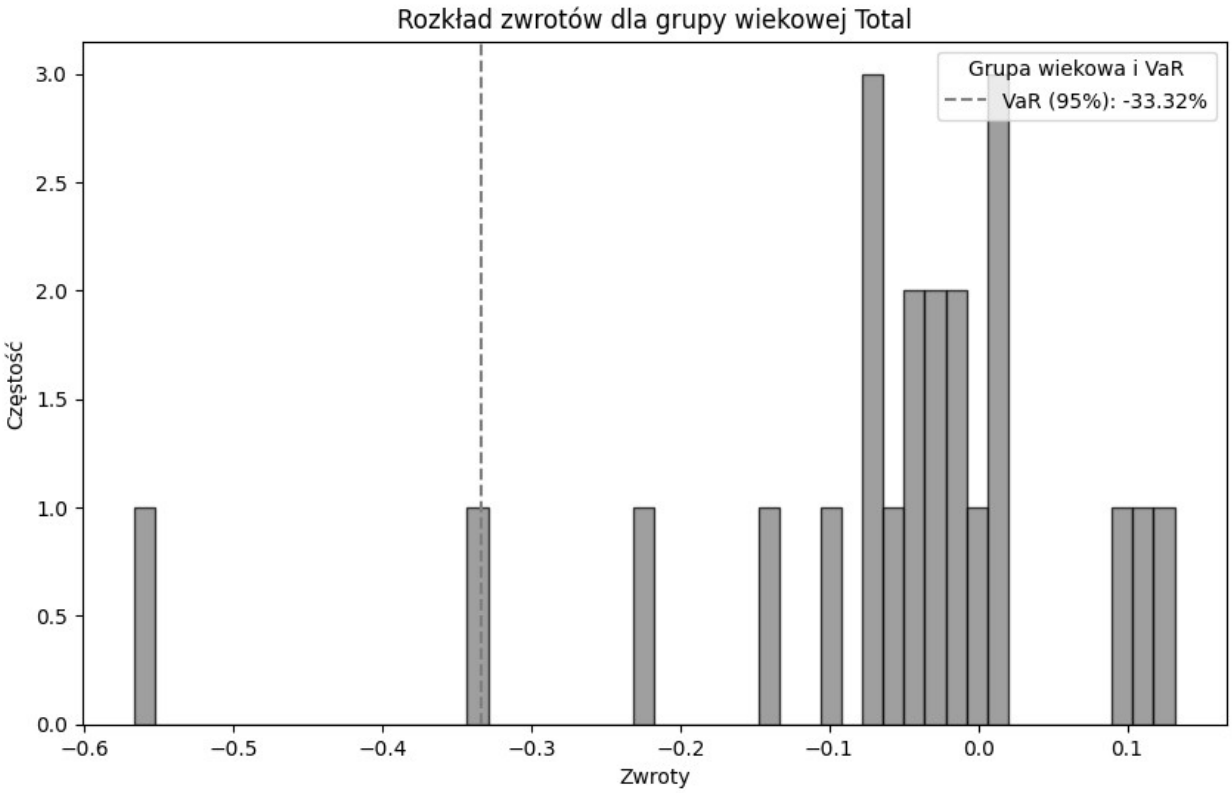

Rozkład zwrotów dla grupy wiekowej 25-39
Grupa wiekowa i VaR
VaR (95%): -32.48%
Częstość
Zwroty
0
1
2
3
4
5
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2


Rozkład zwrotów dla grupy wiekowej 40-54
Grupa wiekowa i VaR
VaR (95%): -35.38%
Częstość
Zwroty
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1


Rozkład zwrotów dla grupy wiekowej 55-64
Grupa wiekowa i VaR
VaR (95%): -34.73%
Częstość
Zwroty
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1


Rozkład zwrotów dla grupy wiekowej 65-79
Grupa wiekowa i VaR
VaR (95%): -34.05%
Częstość
Zwroty
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−0.4
−0.2
0.0
0.2


Rozkład zwrotów dla grupy wiekowej 80+
Grupa wiekowa i VaR
VaR (95%): -31.77%
Częstość
Zwroty
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−0.6
−0.4
−0.2
0.0
0.2
0.4


Rozkład zwrotów dla grupy wiekowej Total
Grupa wiekowa i VaR
VaR (95%): -33.32%
Częstość
Zwroty
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1