Analiza wariancji wskaźnika siły względnej u zawodniczek trójboju siłowego

Celem niniejszej analizy jest zbadanie, czy istnieją istotne statystycznie różnice w wybranym wskaźniku siły względnej pomiędzy grupami wiekowymi kobiet startujących w zawodach trójboju siłowego. Do badania zastosowano **analizę wariancji (ANOVA)** – jedno z podstawowych narzędzi statystycznych umożliwiających porównanie średnich wartości w więcej niż dwóch grupach.

Dane wykorzystane w analizie pochodzą ze zbioru wyników zawodników trójboju siłowego. W celu zwiększenia jednorodności próby oraz ograniczenia wpływu zmiennych zakłócających, uwzględniono jedynie te rekordy, które spełniały następujące kryteria:

- zawodniczka była **kobieta**,
- start odbywał się w formule **Raw** (bez wyposażenia wspomagającego),
- start miał miejsce w dywizji **"Pro Open"**.

Na podstawie tak przefiltrowanego zbioru danych obliczono nową zmienną: **iloraz wyniku w przysiadzie (squat) do wyniku w wyciskaniu na ławce (bench press)**. Wskaźnik ten może odzwierciedlać względne predyspozycje zawodniczek w tych dwóch konkurencjach i służyć jako miara proporcji siły kończyn dolnych do górnych.

Wszystkie zawodniczki zostały następnie przypisane do równolicznych grup wiekowych, co umożliwia porównanie wskaźnika między grupami wiekowymi za pomocą jednoczynnikowej analizy wariancji (ANOVA).

Import bibliotek, wczytanie i wstępna obróbka danych

```python
import pandas as pd
from scipy.stats import shapiro
from scipy.stats import levene
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chisquare

# 1. Wczytanie pliku CSV
df = pd.read_csv('Powerlifters.csv')

# 2. Filtrowanie
df = df[
    (df['Sex'] == 'F') &
    (df['Equipment'] == 'Raw') &
    (df['Division'] == 'Pro Open')
].copy()

# 3. Tworzenie kolumny STBR (squat to bench ratio)
df['STBR'] = df['Best3SquatKg'] / df['Best3BenchKg']

# 4. Usuwanie rekordów, gdzie STBR lub Age są puste lub Best3BenchKg == 0
df = df[
    df['STBR'].notna() &
    df['Age'].notna() &
    df['Best3SquatKg'].notna() &
    df['Best3BenchKg'].notna() &
    (df['Best3BenchKg'] != 0)
].copy()
```

```python
# 5. Grupowanie po imieniu zawodnika i obliczanie średniego STBR i wieku
df_clean = df.groupby('Name').agg({
    'STBR': 'mean',
    'Age': 'mean'
}).reset_index()

# 6. Grupowanie wiekowe
g1 = 27
g2 = 32

def przypisz_grupe_wiekowa(wiek):  1 usage
    if wiek <= g1:
        return 'Młodsi'
    elif wiek <= g2:
        return 'Średni'
    else:
        return 'Starsi'

df_clean['GrupaWiekowa'] = df_clean['Age'].apply(przypisz_grupe_wiekowa)

# 7. Podział na grupy
df1 = df_clean[df_clean['GrupaWiekowa'] == 'Młodsi']
df2 = df_clean[df_clean['GrupaWiekowa'] == 'Średni']
df3 = df_clean[df_clean['GrupaWiekowa'] == 'Starsi']

```

W pierwszym etapie zaimportowano niezbędne biblioteki Pythona, w tym pandas do operacji na danych, scipy.stats do testów statystycznych oraz seaborn i matplotlib do wizualizacji.

Następnie wczytano zbiór danych Powerlifters.csv, zawierający informacje o wynikach zawodników trójboju siłowego. Zbiór został przefiltrowany tak, aby pozostały jedynie rekordy spełniające następujące kryteria:

- płeć: **kobieta**,
- start w formule **Raw** (bez specjalistycznego wyposażenia),
- dywizja: **Pro Open**.

Dla każdej zawodniczki obliczono wskaźnik **STBR (Squat To Bench Ratio)**, będący ilorazem najlepszego wyniku w przysiadzie do najlepszego wyniku w wyciskaniu na klatkę. Następnie usunięto rekordy z brakującymi wartościami oraz przypadki, w których wynik w wyciskaniu wynosił zero, aby uniknąć dzielenia przez zero.

Dane zostały zagregowane tak, aby dla każdej zawodniczki uzyskać pojedynczy rekord zawierający średni wiek oraz średni wskaźnik STBR. Na tej podstawie dokonano podziału na trzy grupy wiekowe:

- **Młodsi**: zawodniczki do 27. roku życia,
- **Średni**: zawodniczki w wieku od 28 do 32 lat,
- **Starsi**: zawodniczki powyżej 32. roku życia.

Dla każdej grupy policzono liczebność, co stanowi podstawę dalszej analizy.

Sprawdzenie założeń analizy wariancji

1. Więcej niż dwie grupy do porównania

Analiza obejmuje trzy grupy wiekowe: *Młodsi*, *Średni* i *Starsi*.

```
# 7. Podział na grupy
df1 = df_clean[df_clean['GrupaWiekowa'] == 'Młodsi']
df2 = df_clean[df_clean['GrupaWiekowa'] == 'Średni']
df3 = df_clean[df_clean['GrupaWiekowa'] == 'Starsi']
```

2. Zmienna zależna na skali ilościowej

Analizowaną zmienną jest wskaźnik **STBR (Squat To Bench Ratio)** – wartość ilorazu dwóch wyników liczbowych (w kg).

```
# 3. Tworzenie kolumny STBR (squat to bench ratio)
df['STBR'] = df['Best3SquatKg'] / df['Best3BenchKg']
```

3. Niezależność obserwacji

Każda zawodniczka reprezentowana jest przez pojedynczy rekord (po agregacji danych wg imienia i nazwiska). Obserwacje są niezależne.

```
df_clean = df.groupby('Name').agg({
    'STBR': 'mean',
    'Age': 'mean'
}).reset_index()
```

4. Równoliczność grup

Grupy wiekowe zostały stworzone tak, by były możliwie wyrównane liczebnie.
Zastosowano test chi-kwadrat dobroci dopasowania

```
licznosci = df_clean['GrupaWiekowa'].value_counts().sort_index()
stat, p = chisquare(f_obs=licznosci)

print(f"\nTest chi² dobroci dopasowania:")
print(f"  Statystyka: {round(stat, 4)}")
print(f"  p-value:    {round(p, 4)}")

if p >= 0.05:
    print(" p>=0.05 - Brak podstaw do odrzucenia hipotezy - liczności są zbliżone.")
else:
    print(" p<0.05 - Różnice w liczności grup są istotne - mogą naruszać założenia.")
```

, którego wynik wskazuje:

```
Test chi² dobroci dopasowania:
  Statystyka: 1.5982
  p-value:    0.4497
 p>=0.05 - Brak podstaw do odrzucenia hipotezy - liczności są zbliżone.
```

5. Jednorodność wariancji

Zastosowano test Levene'a, który sprawdza, czy wariancje analizowanej zmiennej są porównywalne w każdej z grup:

```
# Test Levene'a - weryfikacja jednorodności wariancji
print("\nTest Levene'a (H0: wariancje są jednorodne):")

stat, p = levene( *samples: df1['STBR'], df2['STBR'], df3['STBR'])

print(f"Statystyka testu = {round(stat, 4)}, p-value = {round(p, 4)}")

if p >= 0.05:
    print(" p>=0.05 - Brak podstaw do odrzucenia H₀ - wariancje można uznać za jednorodne.")
else:
    print(" p<0.05 - Odrzucamy H₀ - istnieją istotne różnice w wariancjach między grupami.")
```

, którego wynik wskazuje:

```
Test Levene'a (H0: wariancje są jednorodne):
Statystyka testu = 1.7024, p-value = 0.183
 p>=0.05 - Brak podstaw do odrzucenia H₀ - wariancje można uznać za jednorodne.
```

6. Normalność rozkładu (ważna zwłaszcza przy n < 30)

-W każdej z grup wykonano test Shapiro-Wilka:

```
# Shapiro-Wilk
print("\nTest Shapiro-Wilka (H0: rozkład normalny):")

for grupa, dane in zip(['Młodsi', 'Średni', 'Starsi'], [df1, df2, df3]):
    stat, p = shapiro(dane['STBR'])
    print(f"{grupa}: statystyka = {round(stat, 4)}, p-value = {round(p, 4)}")

    if p >= 0.05:
        print(" p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.\n")
    else:
        print(" p<0.05 - Odrzucamy H₀ - rozkład odbiega od normalnego.\n")
```

,którego wynik wskazuje:

```
Test Shapiro-Wilka (H0: rozkład normalny):
Młodsi: statystyka = 0.9972, p-value = 0.9518
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.

Średni: statystyka = 0.9934, p-value = 0.4495
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.

Starsi: statystyka = 0.9892, p-value = 0.0989
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.
```

-W każdej z grup wykonano test Kolmogorova-Smirnova.

```
# Kolmogorov-Smirnov
print("\nTest Kolmogorova-Smirnova:")
for grupa, dane in zip(['Młodsi', 'Średni', 'Starsi'], [df1, df2, df3]):
    z = zscore(dane['STBR'])
    stat, p = kstest(z, cdf: 'norm')
    print(f"{grupa}: stat = {round(stat, 4)}, p = {round(p, 4)}")
    if p >= 0.05:
        print(" p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.\n")
    else:
        print(" p<0.05 - Odrzucamy H₀ - rozkład odbiega od normalnego.\n")
```

,którego wynik wskazuje:

```
Test Kolmogorova-Smirnova:
Młodsi: stat = 0.0376, p = 0.8715
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.

Średni: stat = 0.0568, p = 0.4716
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.

Starsi: stat = 0.0606, p = 0.3788
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.
```

-W każdej z grup wykonano test Andersona-Darlinga.

```
# Anderson-Darling
print("\nTest Andersona-Darlinga:")
for grupa, dane in zip(['Młodsi', 'Średni', 'Starsi'], [df1, df2, df3]):
    wynik = anderson(dane['STBR'], dist='norm')
    stat = wynik.statistic
    granica = wynik.critical_values[2]  # poziom 5%
    print(f"{grupa}: stat = {round(stat, 4)}, granica (5%) = {round(granica, 4)}")
    if p >= 0.05:
        print(" p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.\n")
    else:
        print(" p<0.05 - Odrzucamy H₀ - rozkład odbiega od normalnego.\n")
```

,którego wynik wskazuje:

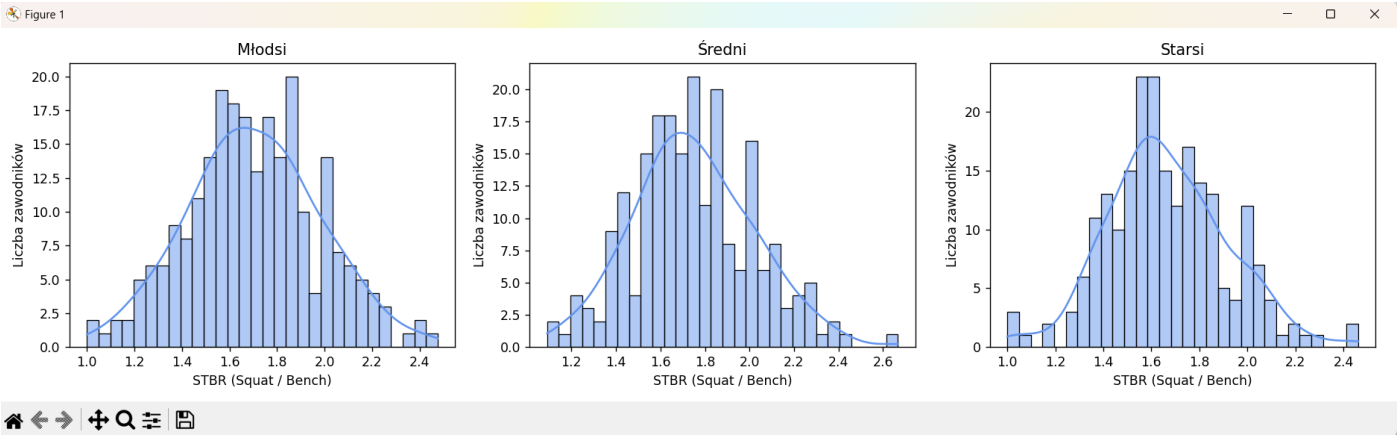
```
Test Andersona-Darlinga:
Młodsi: stat = 0.1564, granica (5%) = 0.774
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.

Średni: stat = 0.42, granica (5%) = 0.773
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.

Starsi: stat = 0.7453, granica (5%) = 0.773
 p>=0.05 - Brak podstaw do odrzucenia H₀ - rozkład może być normalny.
```

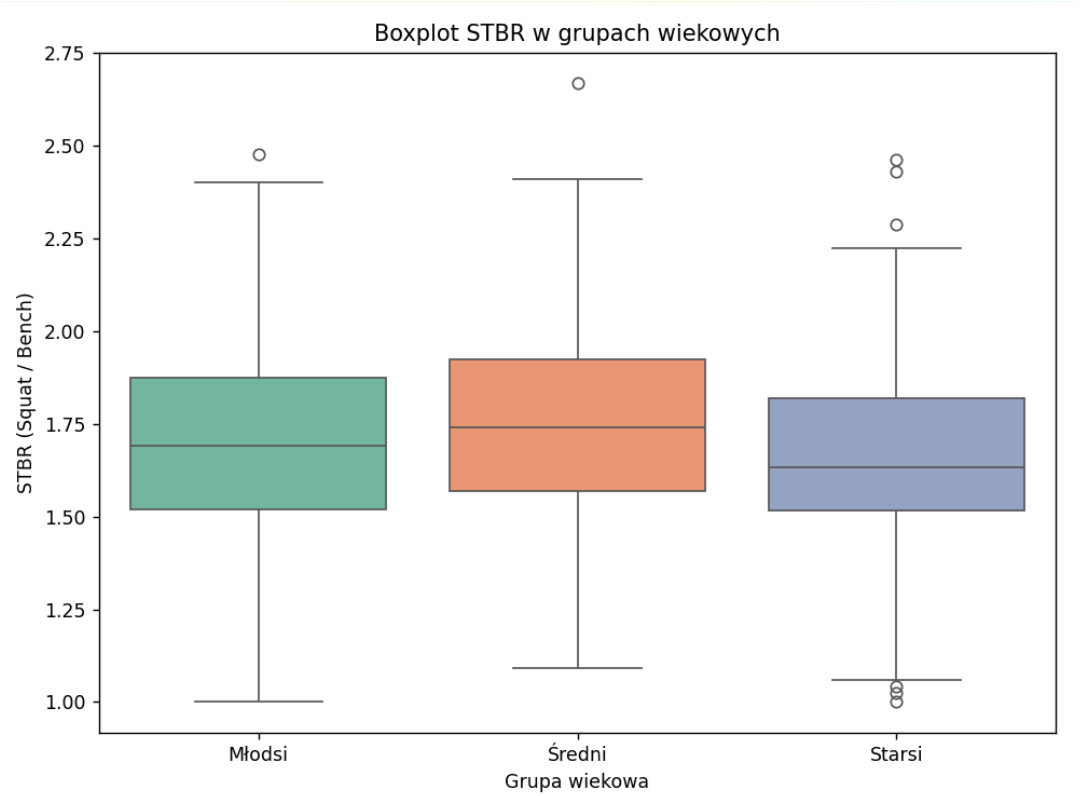
Histogramy z nałożonymi krzywymi gęstości

Każdy z trzech histogramów odpowiada jednej grupie wiekowej. Wszystkie wykresy pokazują kształt zbliżony do rozkładu normalnego, co wspiera wnioski z testu Shapiro-Wilka.



Boxplot STBR w podziale na grupy wiekowe

Boxploty umożliwiają ocenę symetrii rozkładu oraz obecności wartości odstających. W każdej z grup dane rozkładają się stosunkowo równomiernie wokół mediany, bez znaczącego skośności.



Założenie spełnione – zarówno test statystyczny, jak i wizualizacje wskazują na brak istotnych odchyleń od rozkładu normalnego.