Dokumentacja Lab3

Analiza wstępna danych

Charakterystyka zbioru danych:

- **Liczba obserwacji:** 4739
- **Liczba zmiennych:** 15

Dane obejmują zarówno zmienne numeryczne, jak i kategoryczne:

- **Zmienne numeryczne:**
  - rownames (identyfikator)
  - score (zmienna docelowa)
  - unemp
  - wage
  - distance
  - tuition
  - education
- **Zmienne kategoryczne:**
  - gender
  - ethnicity
  - fcollege
  - mcollege
  - home
  - urban
  - income
  - region

Braki danych:

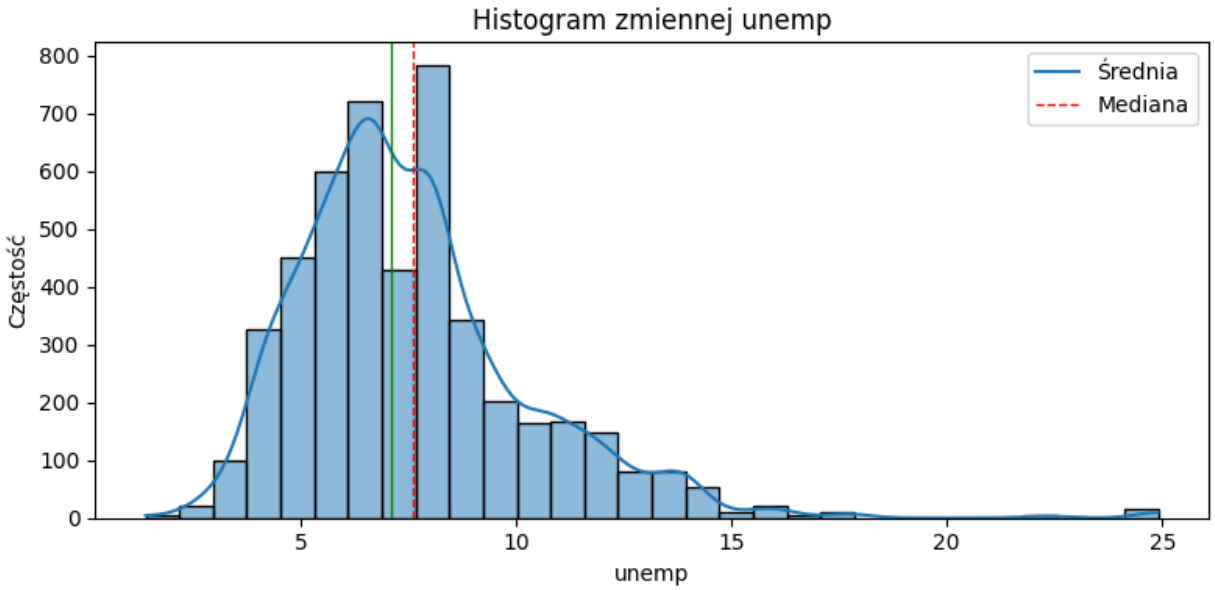
- W zbiorze danych **nie występują wartości brakujące**; wszystkie kolumny są kompletne.
- Ewentualne braki mogą być reprezentowane przez wartość **0** w zmiennych numerycznych.

Analiza statystyczna

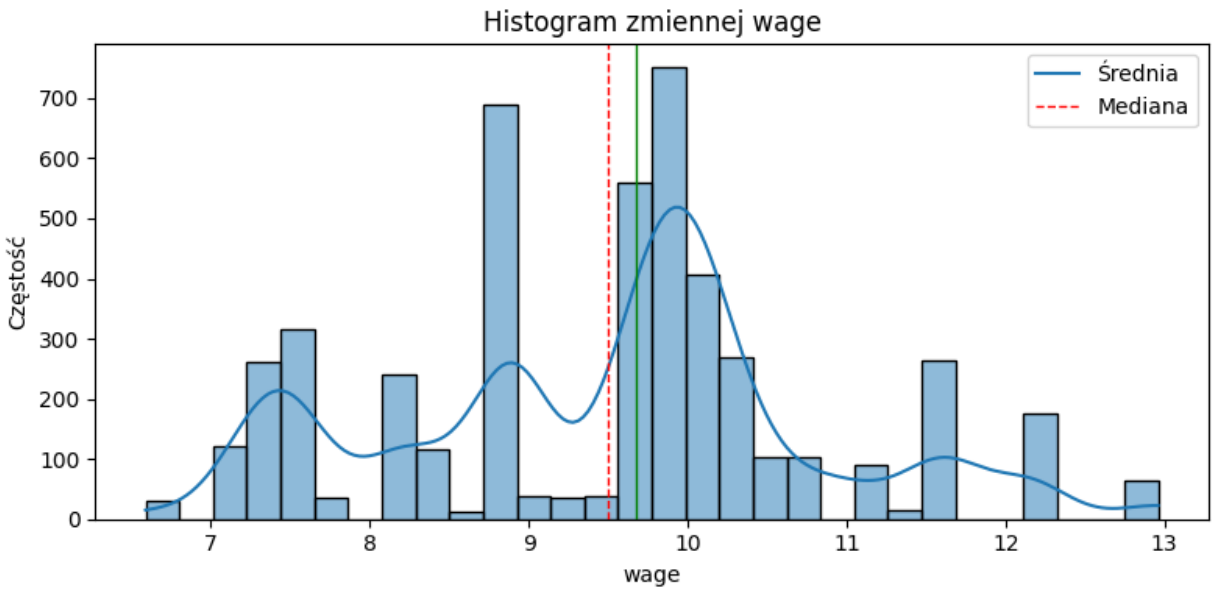
Zmienne numeryczne:

1. **Unemployment (unemp):**

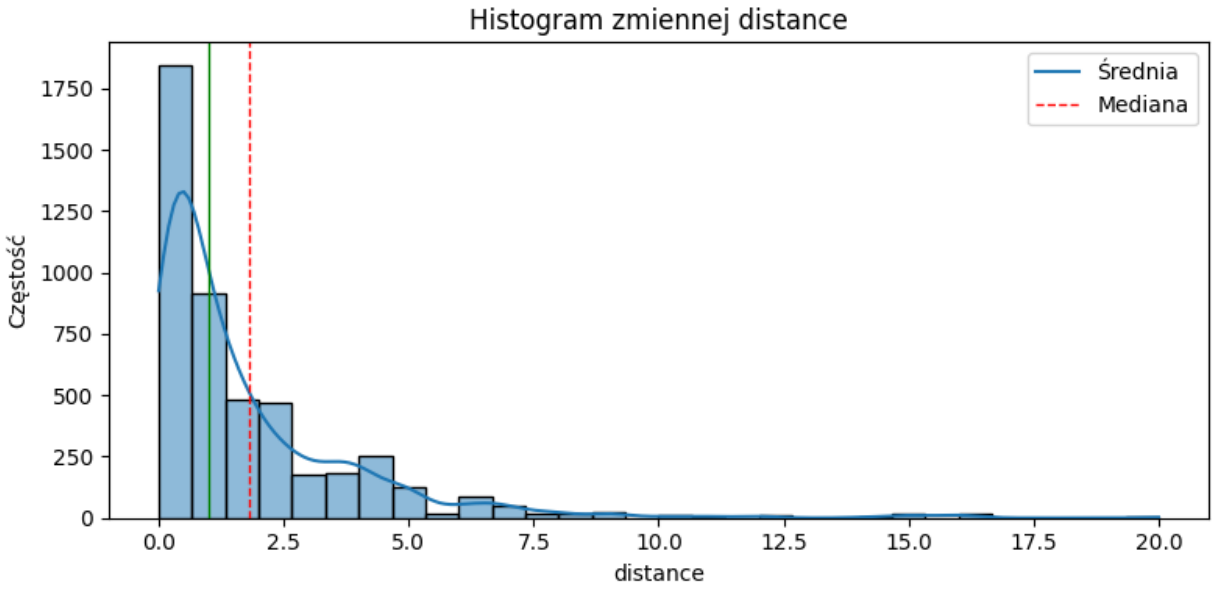
- Średnia: **7.6**
- Mediana: **7.1**
- Zakres: **1.4 – 24.9**



2. **Wynagrodzenie (wage):**
   - Średnia: **9.5**
   - Mediana: **9.7**
   - Zakres: **6.59 – 12.96**

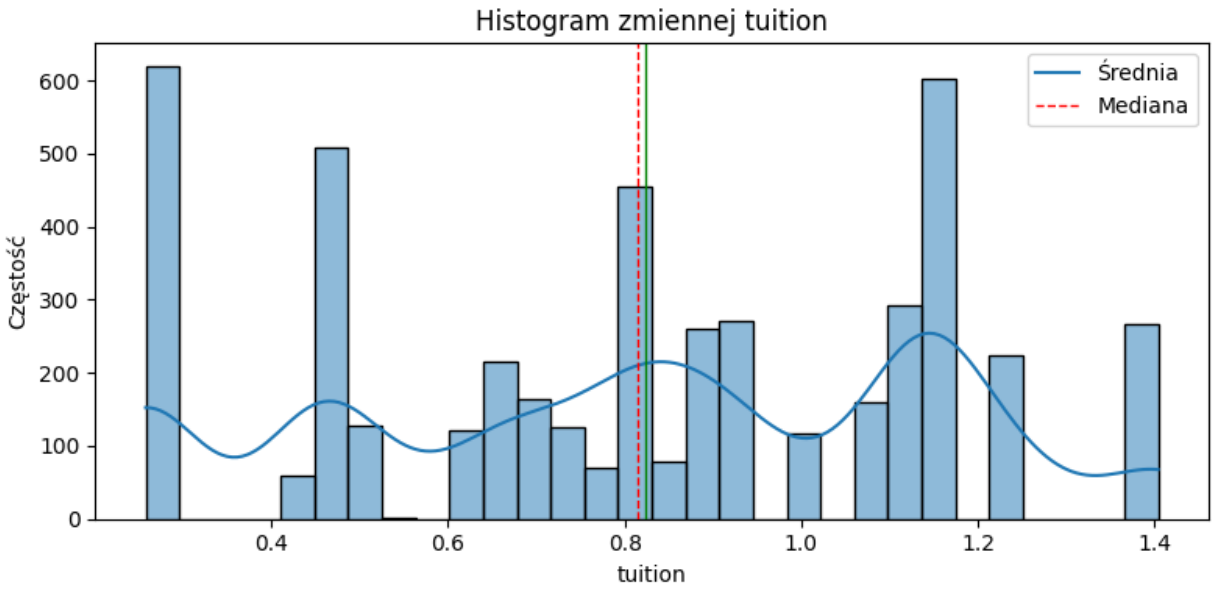


3. **Dystans do uczelni (distance):**
    - Średnia: **1.8**
    - Mediana: **1.0**
    - Zakres: **0.0 – 20.0**

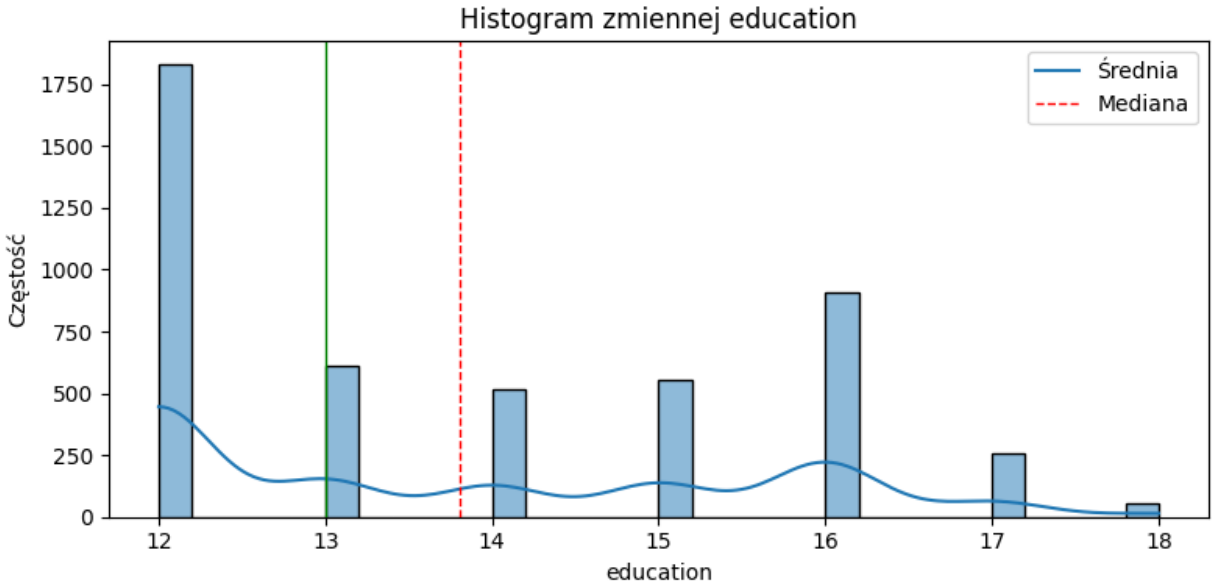


4. **Czesne (tuition):**
    - Średnia: **0.81**

- Mediana: **0.82**
- Zakres: **0.26 – 1.40**

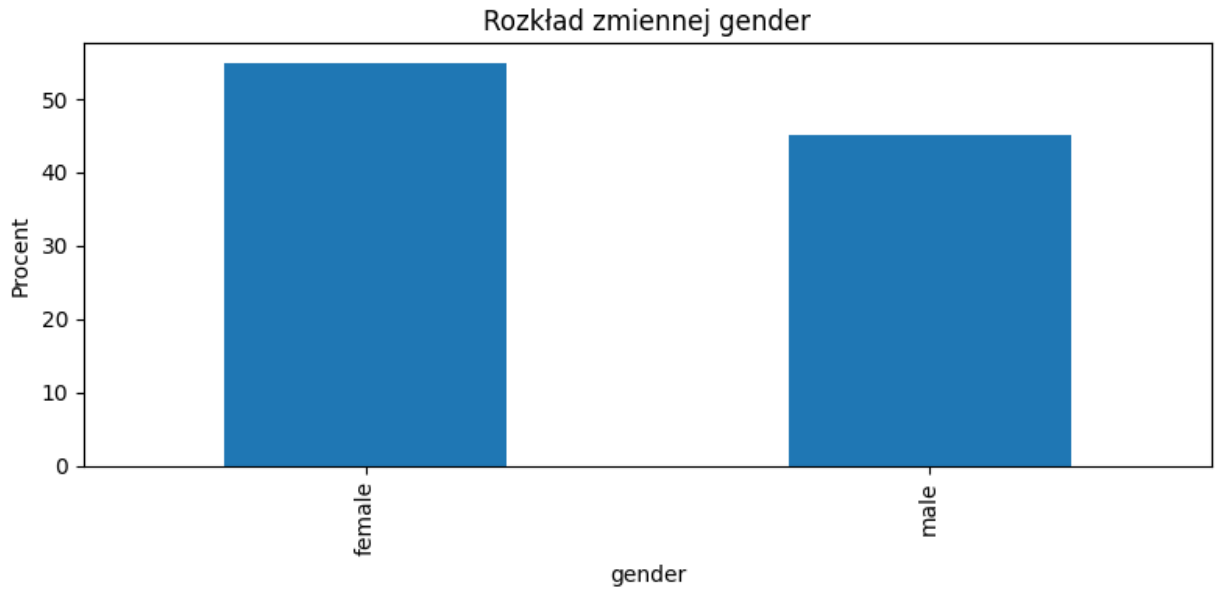


5. **Edukacja (education):**
   - Średnia: **13.8**
   - Mediana: **13**
   - Zakres: **12 – 18**

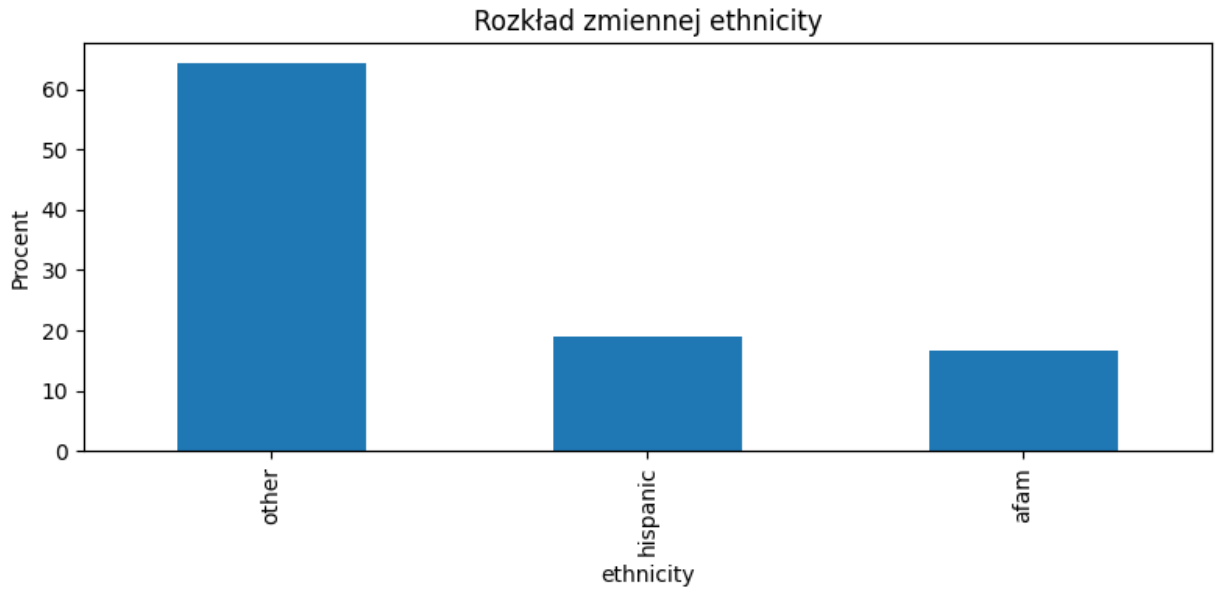


Zmienne kategoryczne:

1. **Płeć (gender):**
    - Kategorie: **male**, **female**

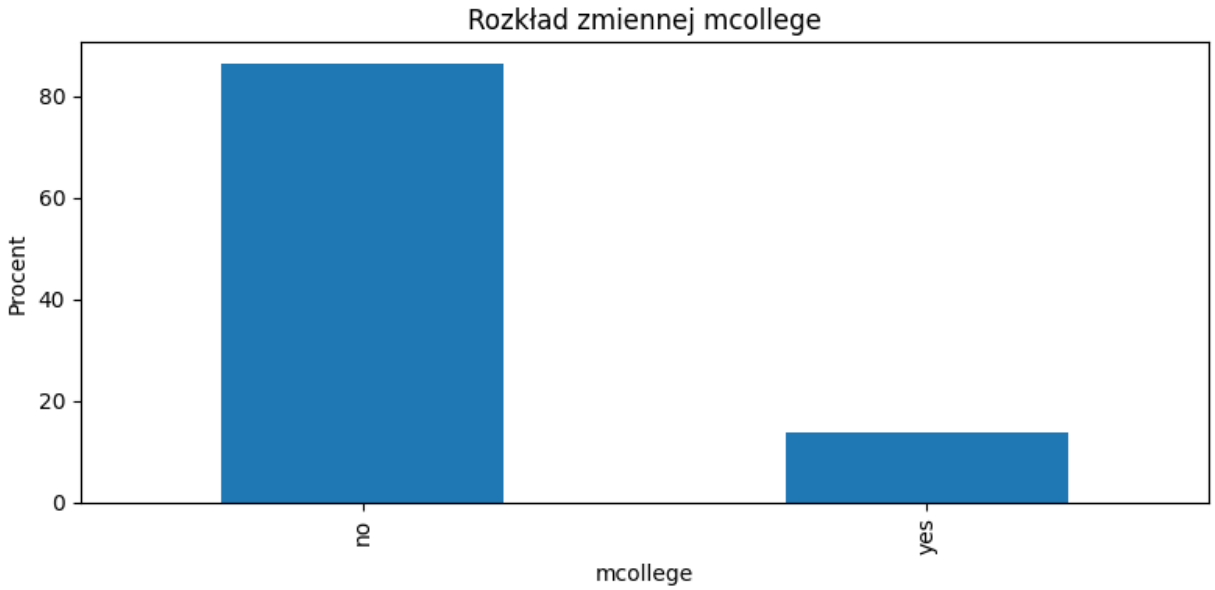


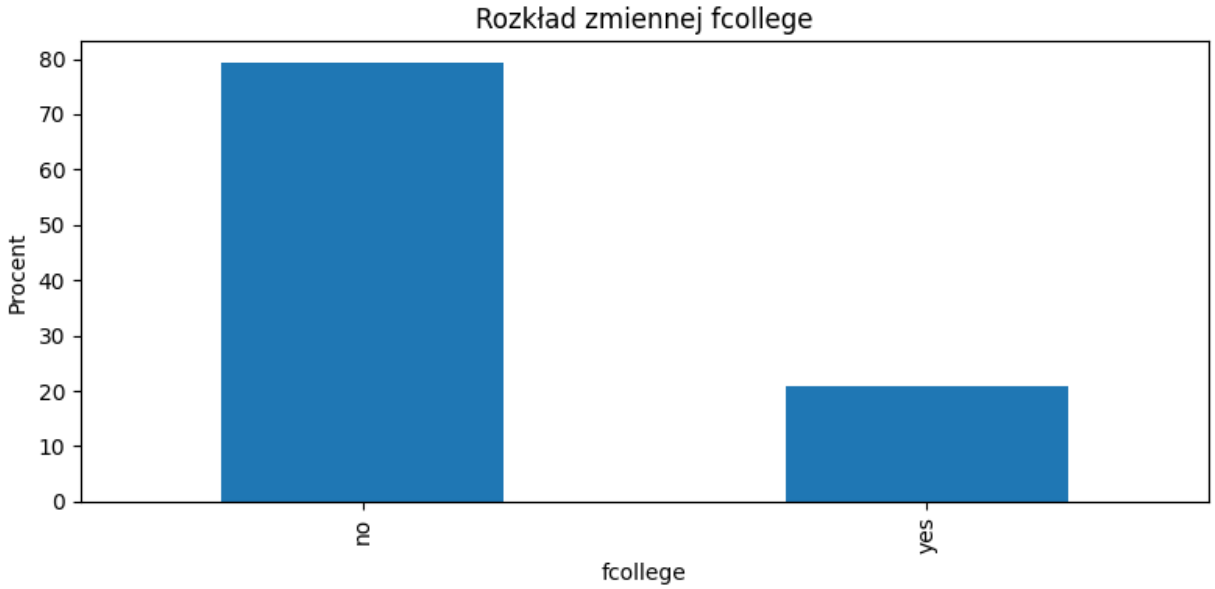
2. **Etniczność (ethnicity):**
    - Kategorie: **afam** (Afroamerykanin), **hispanic**, **other**



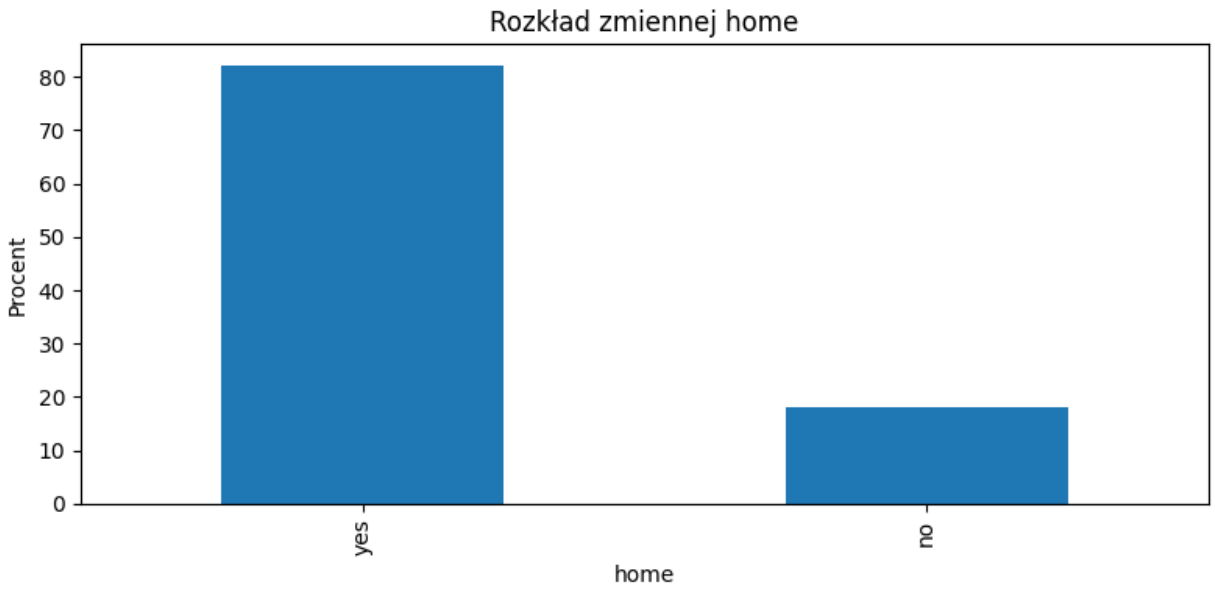
3. **Wykształcenie rodziców (fcollege, mcollege):**

- Kategorie: **yes**, **no**

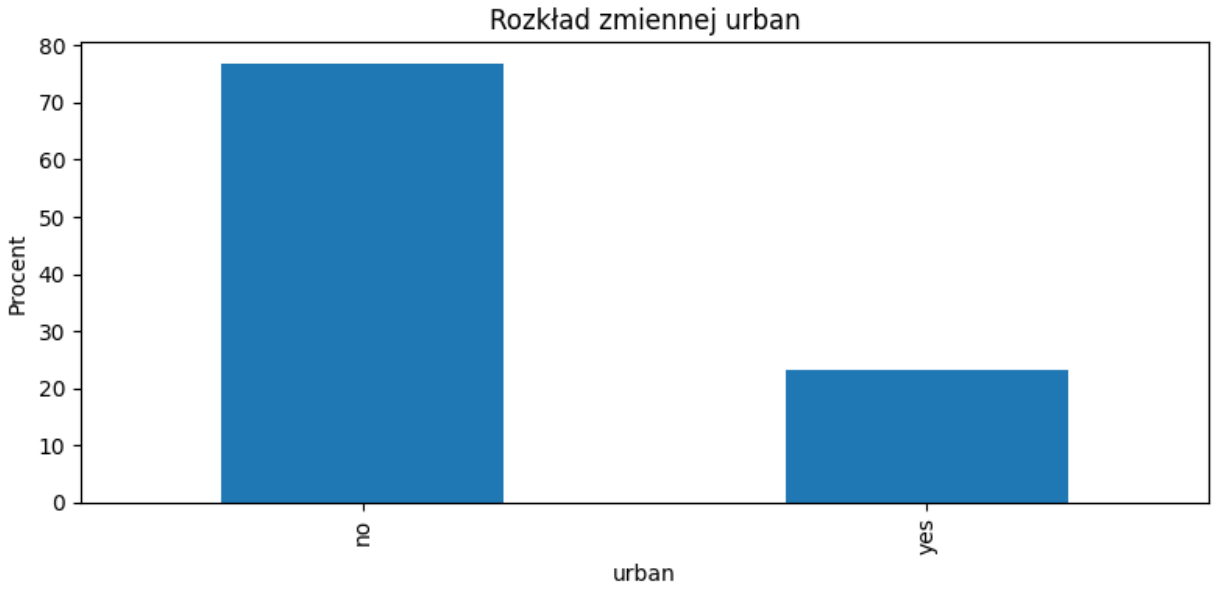




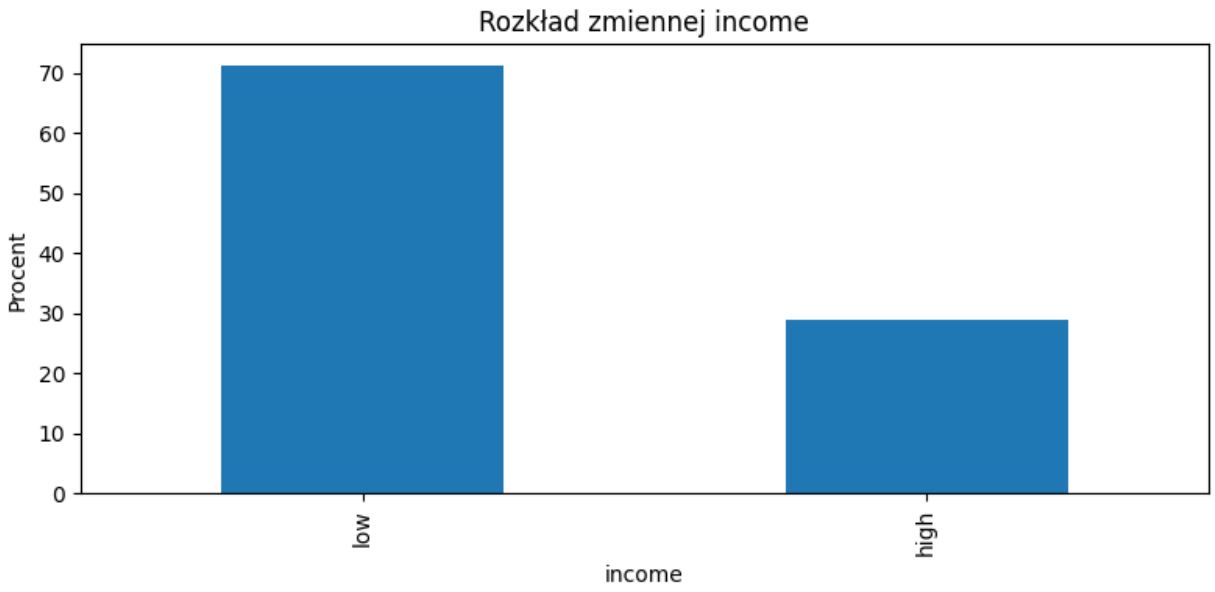
4. **Mieszkanie w domu (home):**
    - Kategorie: **yes**, **no**



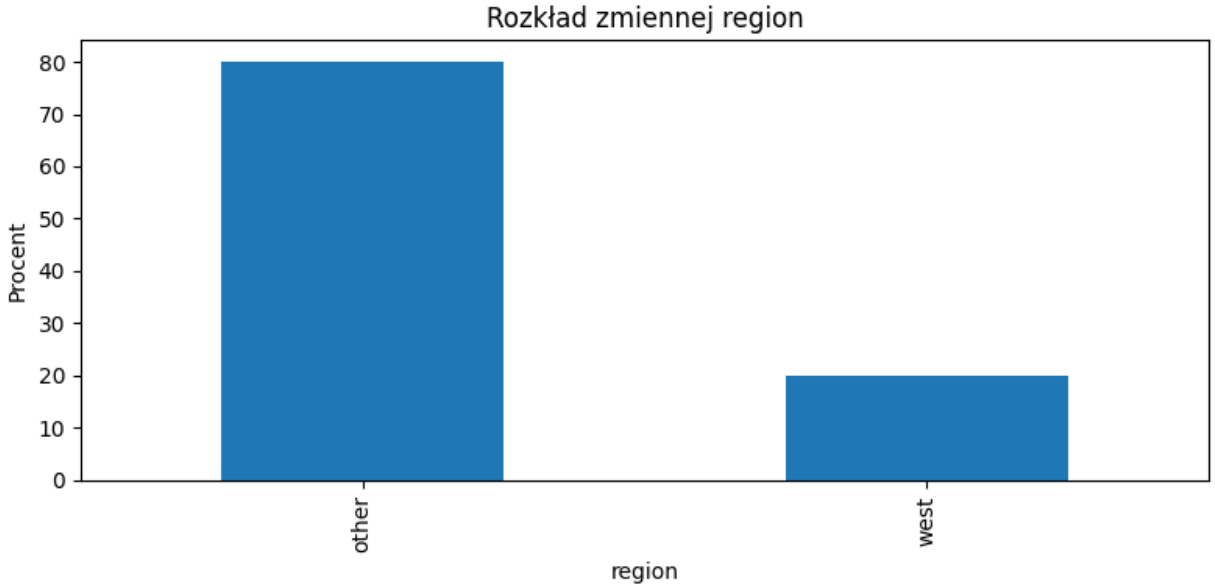
5. **Obszar zamieszkania (urban):**
    - Kategorie: **yes**, **no**



6. **Dochód rodziny (income):**
    - Kategorie: **high**, **low**

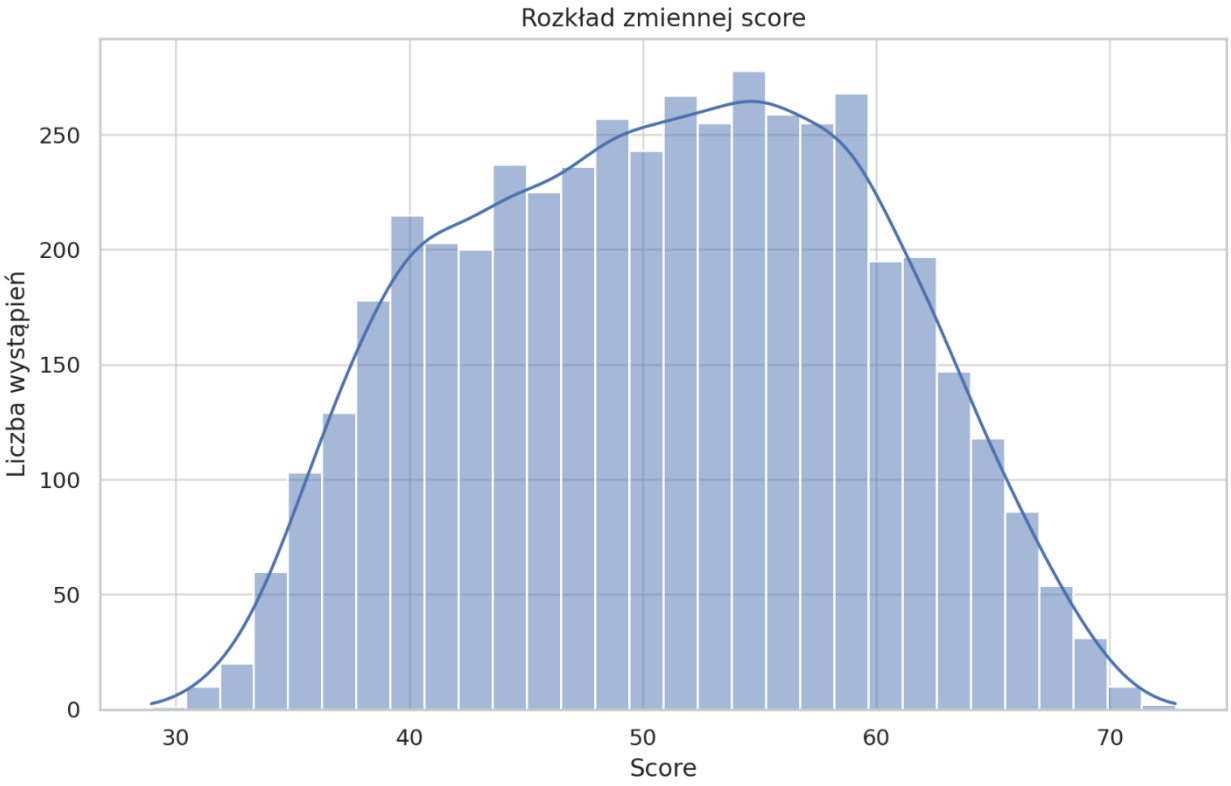


7. **Region (region):**
    - Kategorie: **west**, **other**

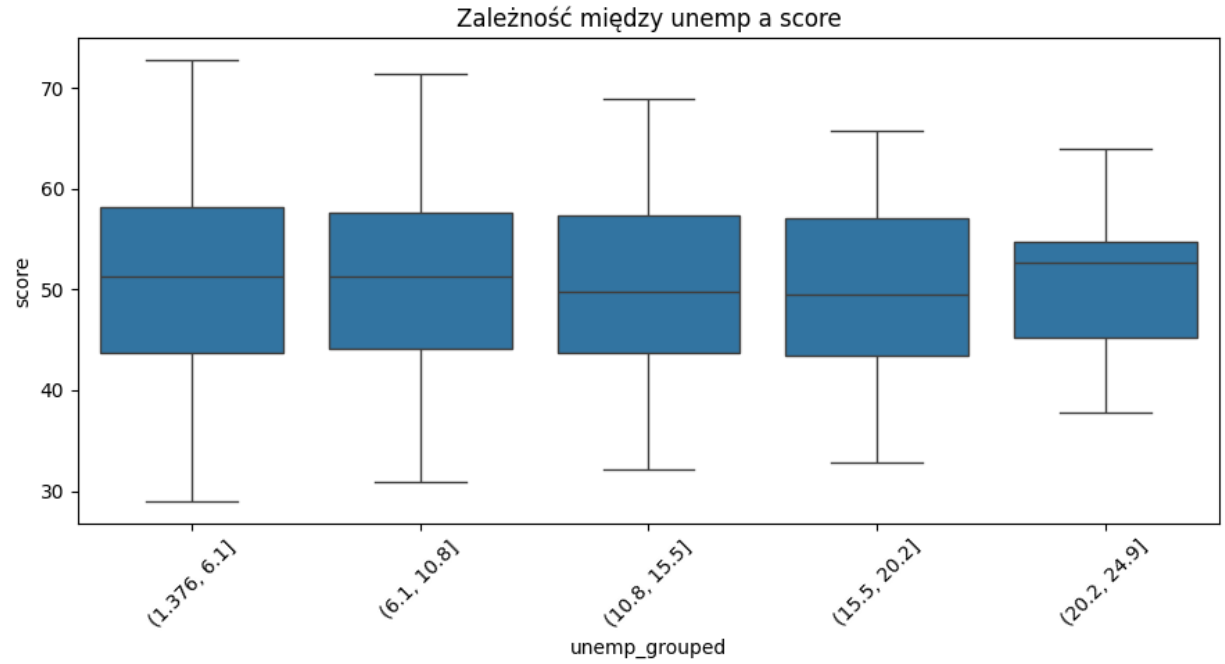


Wizualizacja zależności między danymi

- **Rozkład zmiennej score** jest zbliżony do normalnego, z pewnymi asymetriami i wartościami odstającymi. Większość wyników mieści się w zakresie **40–60**.

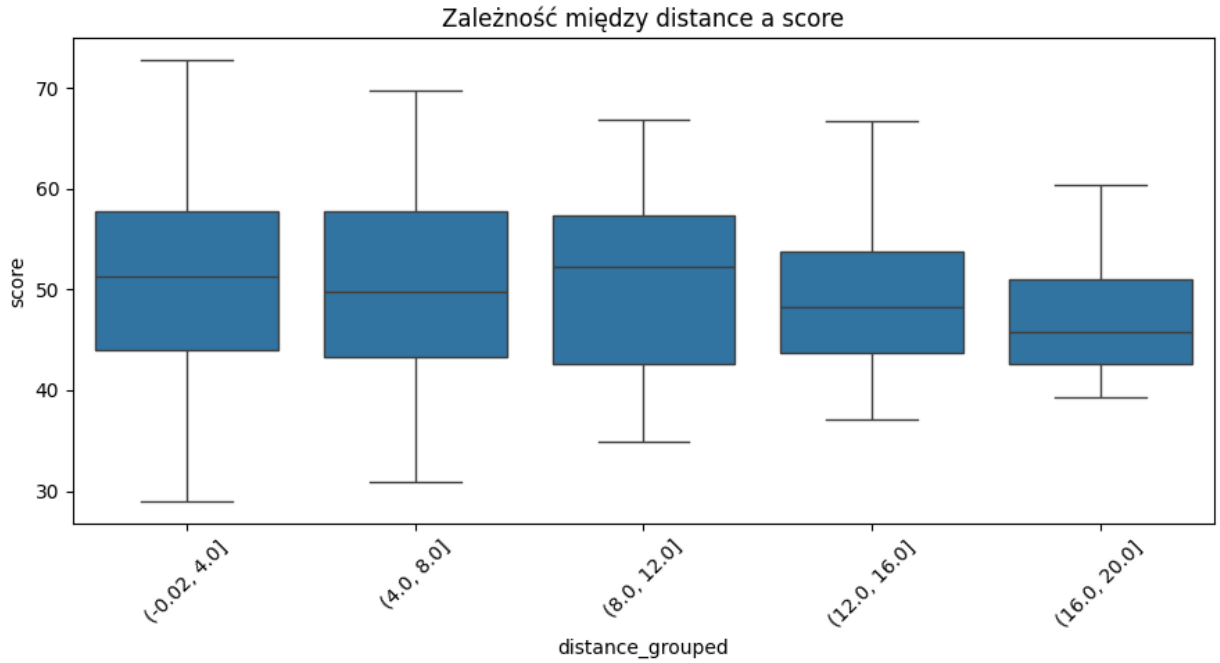


- **Bezrobocie (unemp):** Brak jednoznacznej korelacji między poziomem bezrobocia a wartością score.

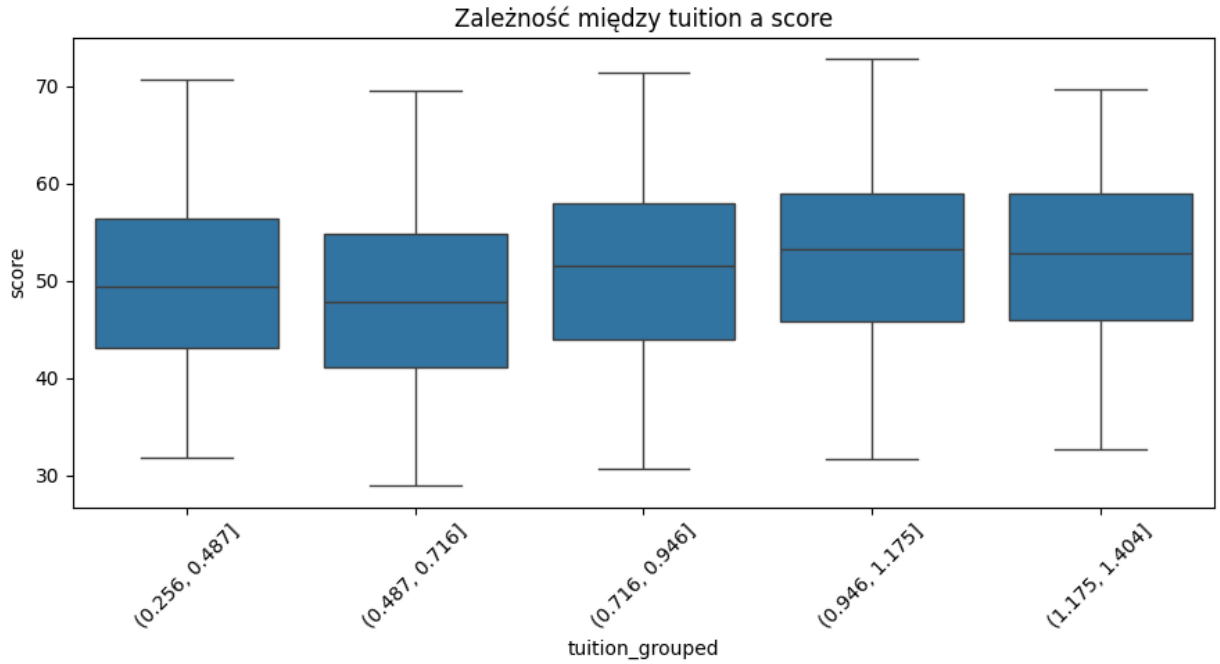


- **Wynagrodzenie (wage):** Umiarkowana, nieliniowa zależność między wyższym wynagrodzeniem a wyższymi wynikami score.

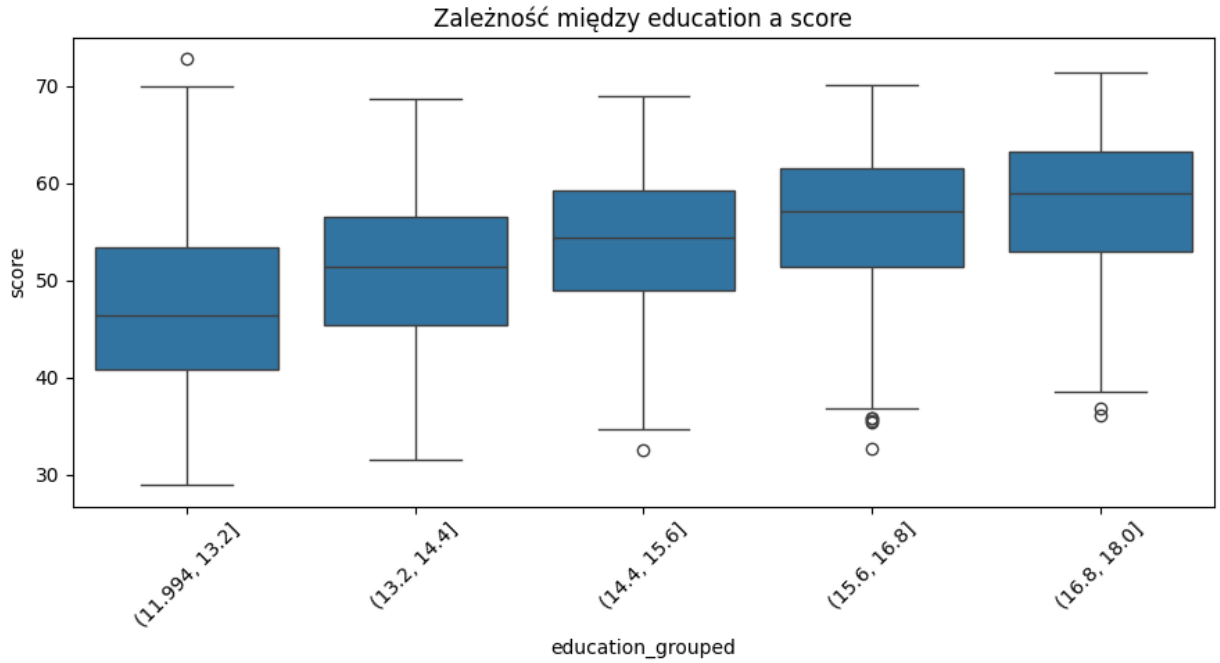
- **Dystans do uczelni (distance):** Ogólnie, większy dystans obniża score, ale powyżej **2.5 km** obserwuje się większą zmienność wyników.



- **Czesne (tuition):** Nieznaczna pozytywna korelacja między wyższym czesnym a wynikiem score.

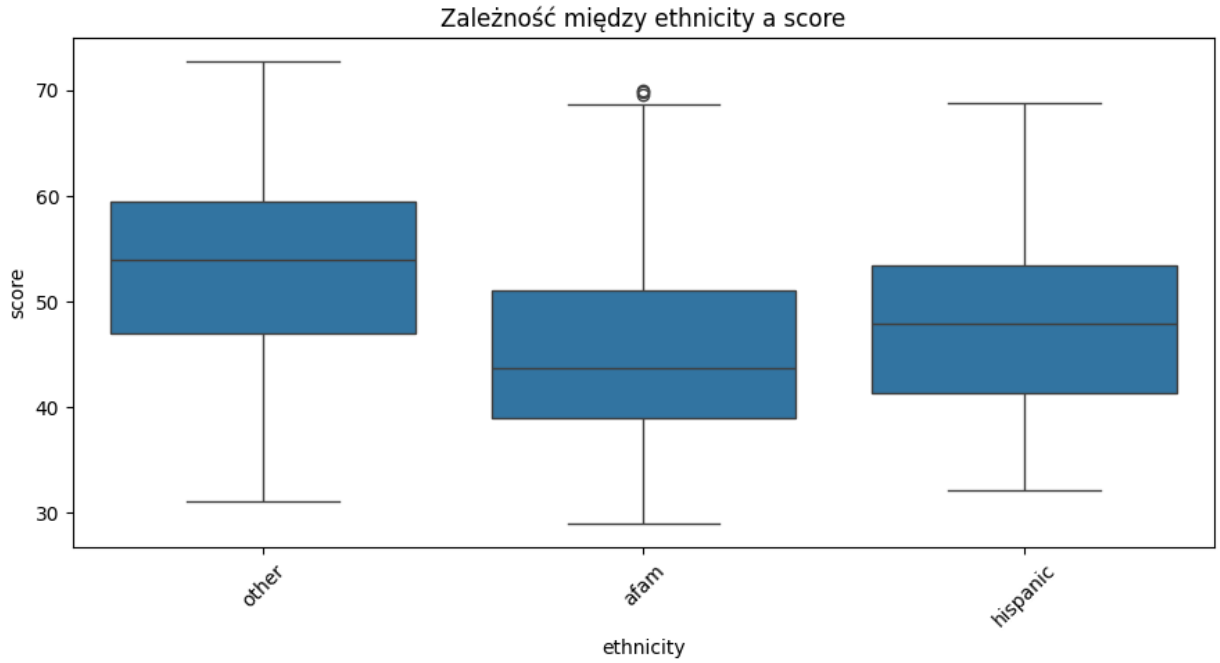


- **Edukacja (education):** Wyższy poziom edukacji koreluje z wyższymi wartościami score.

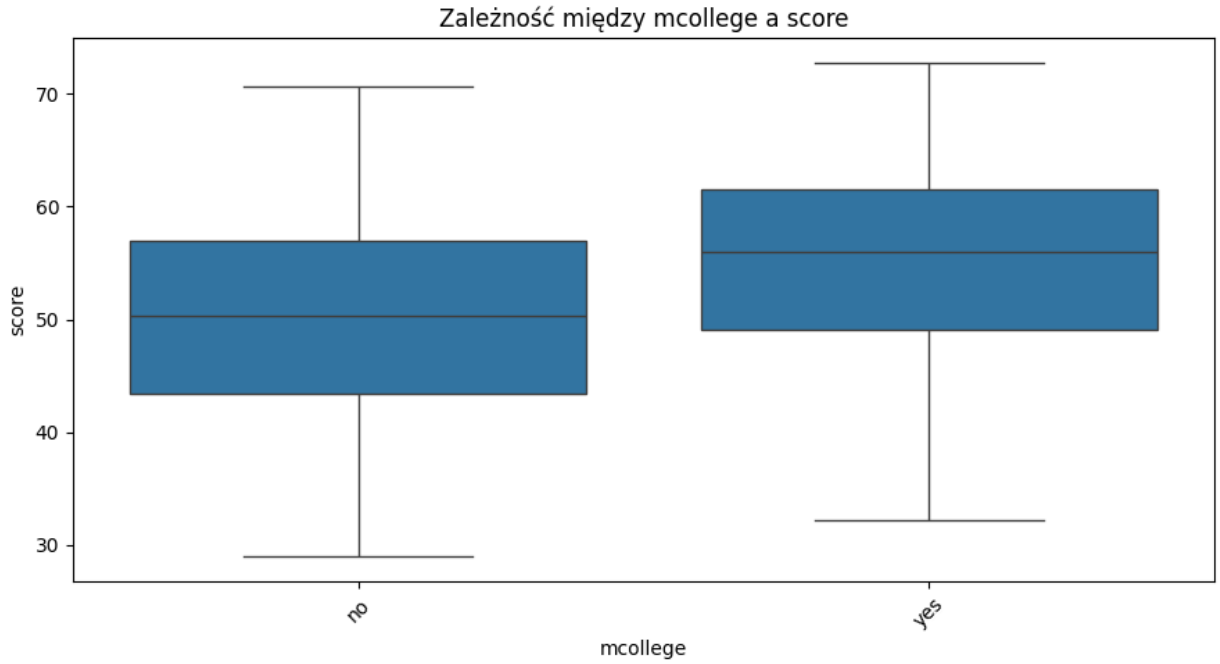


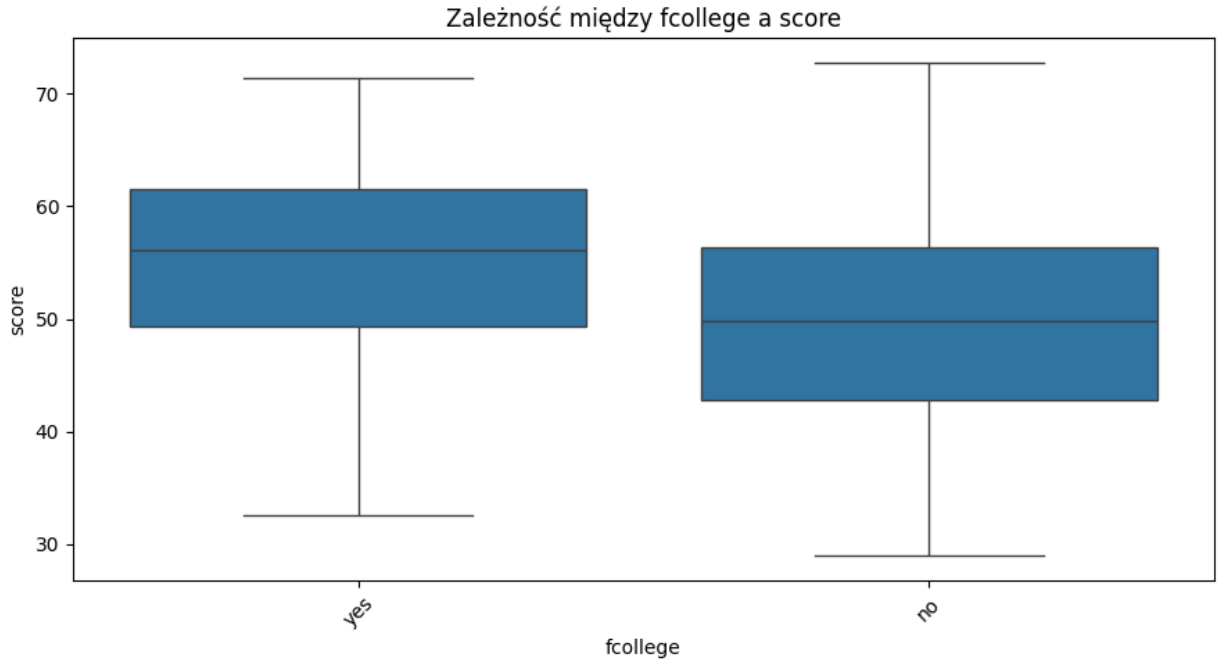
- **Płeć (gender):** Kobiety osiągają nieco wyższe wyniki score w porównaniu do mężczyzn.

- **Etniczność (ethnicity):** Afroamerykanie mają tendencję do niższych wyników score w porównaniu z innymi grupami etnicznymi.

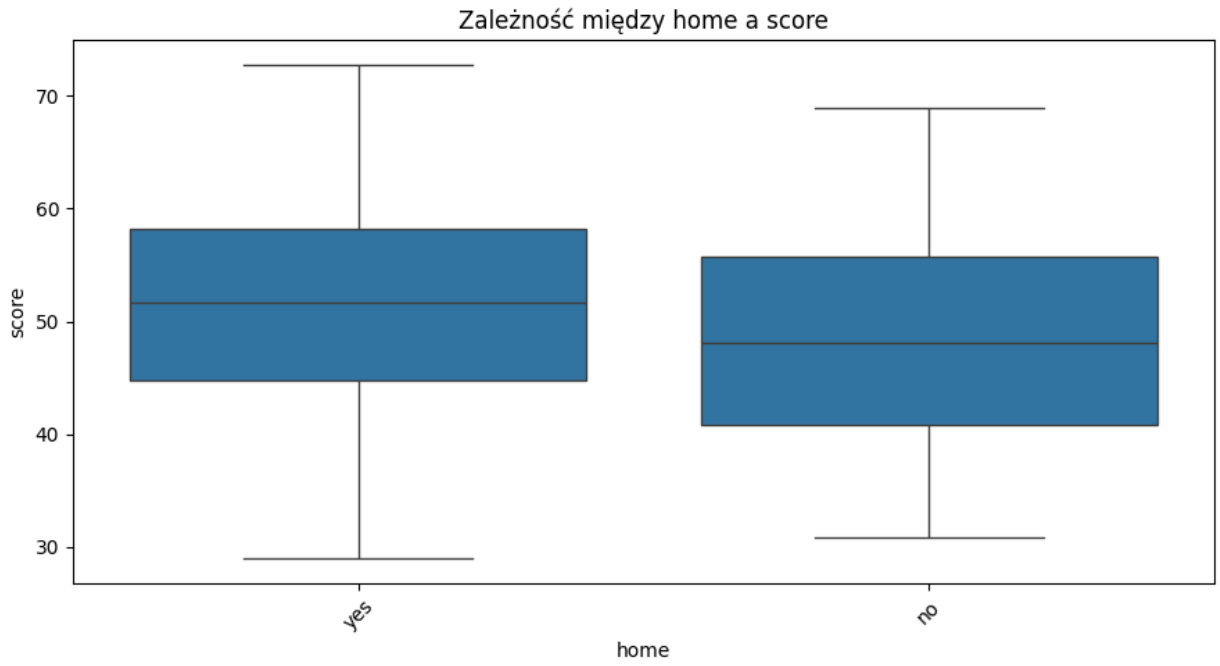


- **Wykształcenie rodziców (fcollege, mcollege):** Wykształceni rodzice pozytywnie wpływają na wyniki score.





- **Mieszkanie w domu (home):** Studenci mieszkający w domu często osiągają niższe wyniki score.



- **Obszar zamieszkania (urban):** Mieszkanie w mieście nieznacznie podnosi wynik score.

- **Dochód rodziny (income):** Wyższy dochód rodziny jest skorelowany z wyższym score.

- **Region (region):** Studenci z regionu **west** osiągają wyższe wyniki score niż z pozostałych regionów.

Inżynieria cech i przygotowanie danych

Aby przygotować dane do modelowania, zastosowano następujące kroki:

- **Imputacja brakujących wartości:**
    - **Zmienne numeryczne:** Uzupełniono średnią (SimpleImputer(strategy='mean')).
    - **Zmienne kategoryczne:** Uzupełniono najczęstszą kategorią (SimpleImputer(strategy='most_frequent')).
- **Standaryzacja zmiennych numerycznych:**
    - Zastosowano **StandardScaler** do przekształcenia zmiennych, aby miały średnią **0** i odchylenie standardowe **1**.
- **Kodowanie zmiennych kategorycznych:**
    - Użyto techniki **one-hot encoding** do przekształcenia kategorii na zmienne binarne.
- **Podział zbioru danych:**
    - Dane podzielono na zbiór **treningowy** (80%) i **testowy** (20%).

Kod odpowiedzialny za inżynierię cech i przygotowanie danych znajduje się w pliku ***data_prediction.py***.

```python
# Wczytanie danych
data = pd.read_csv('CollegeDistance.csv')

# Definicja zmiennych numerycznych i kategorycznych
numerical_features = ['unemp', 'wage', 'distance', 'tuition', 'education']
categorical_features = ['gender', 'ethnicity', 'fcollege', 'mcollege', 'home', 'urban', 'income', 'region']

# Definicja zmiennej docelowej
target_variable = 'score'

# Podział danych na cechy i target
X = data.drop(columns=[target_variable])
y = data[target_variable]

# Podział na zbiór treningowy i testowy
X_train, X_test, y_train, y_test = train_test_split(
    *arrays: X, y, test_size=0.2, random_state=42
)

# Transformacje dla cech numerycznych
numeric_transformer = make_pipeline(
    *steps: KNNImputer(n_neighbors=5),
    MinMaxScaler()
)

# Transformacje dla cech kategorycznych
categorical_transformer = make_pipeline(
    *steps: SimpleImputer(strategy='most_frequent'),
    OneHotEncoder(handle_unknown='ignore')
)

# Połączenie transformacji
preprocessor = make_column_transformer(
    *transformers: (numeric_transformer, numerical_features),
    (categorical_transformer, categorical_features)
)

# Utworzenie pełnego pipeline z modelem
model_pipeline = make_pipeline(
    *steps: preprocessor,
    RandomForestRegressor(random_state=42)
)
```

Wybór modelu predykcyjnego

Ze względu na charakter zadania (przewidywanie zmiennej ciągłej score), problem ten jest problemem regresji. Rozważono kilka modeli:

- **Regresja liniowa:**
    - Szybka i efektywna przy prostych, liniowych zależnościach.
    - Może być niewystarczająca przy skomplikowanych relacjach między zmiennymi.

- **Lasy losowe (Random Forest):**
    - Radzi sobie dobrze z nieliniowymi i złożonymi zależnościami.
    - Odporność na nadmierne dopasowanie dzięki technice baggingu.
- **Gradient Boosting (np. XGBoost, LightGBM):**
    - Zaawansowane algorytmy skuteczne w modelowaniu skomplikowanych zależności.
    - Wymagają więcej zasobów obliczeniowych.
- **Regresja LASSO / Ridge:**
    - Modele liniowe z regularyzacją, pomagające zapobiegać nadmiernemu dopasowaniu.

Wybór modelu Random Forest

Model **Random Forest** został wybrany ze względu na:

- **Elastyczność:** Dobrze radzi sobie z różnymi typami zmiennych (numeryczne i kategoryczne).
- **Zdolność do modelowania złożonych zależności:** Uchwyca nieliniowe relacje między cechami a zmienną docelową.
- **Odporność na nadmierne dopasowanie:** Możliwość regulacji poprzez hiperparametry.
- **Brak założenia liniowości:** Nie wymaga, aby dane były liniowo zależne.

Trenowanie modelu

Po wstępnym wytrenowaniu modelu Random Forest z domyślnymi ustawieniami uzyskano:

- **Zbiór treningowy:**
    - **MSE:** 10.4788
    - **MAE:** 2.5032
    - **RMSE:** 3.2371
    - **R^2:** 0.8615
    - **MAPE:** 5.13%
- **Zbiór testowy:**

- **MSE:** 53.6624
- **MAE:** 5.8536
- **RMSE:** 7.3255
- **R^2:** 0.2924
- **MAPE:** 12.04%

Znaczna różnica między wynikami na zbiorze treningowym a testowym sugeruje **nadmierne dopasowanie** modelu.

Wyjaśnienie metryk:

- **MSE (Mean Squared Error):** Średni błąd kwadratowy między przewidywaniami a rzeczywistymi wartościami.
- **MAE (Mean Absolute Error):** Średnia wartość bezwzględna różnic między przewidywaniami a rzeczywistymi wartościami.
- **RMSE (Root Mean Squared Error):** Pierwiastek kwadratowy z MSE; interpretowany w jednostkach zmiennej docelowej.
- **R^2 (R-squared):** Proporcja wariancji wyjaśnionej przez model.
- **MAPE (Mean Absolute Percentage Error):** Średni procentowy błąd bezwzględny.

Optymalizacja modelu

Aby poprawić wydajność modelu, zastosowano:

- **Walidację krzyżową:**
  - Pomaga w ocenie modelu na różnych podzbiorach danych, zmniejszając ryzyko nadmiernego dopasowania.
- **Tunowanie hiperparametrów (Randomized Search):**
  - Użyto **RandomizedSearchCV** do znalezienia optymalnych wartości hiperparametrów modelu Random Forest.
  - Randomized Search przeszukuje losowo przestrzeń hiperparametrów, co jest efektywne przy dużej liczbie możliwych kombinacji.

Wyniki po optymalizacji:

- **Zbiór testowy:**
  - **MSE:** 49.1919
  - **MAE:** 5.6841

- **RMSE:** 7.0137
- **MAPE:** 11.74%
- **R^2:** 0.3513

Mimo pewnej poprawy, różnice między zbiorem treningowym a testowym nadal są znaczące.

*Kod odpowiedzialny za optymalizację modelu znajduje się w pliku **data_prediction.py**.*

```python
# Utworzenie pełnego pipeline z modelem
model_pipeline = make_pipeline(
    *steps: preprocessor,
    RandomForestRegressor(random_state=42)
)

# Parametry do przeszukania
param_distributions = {
    'randomforestregressor__n_estimators': [100, 200, 300],
    'randomforestregressor__max_depth': [None, 10, 20],
    'randomforestregressor__min_samples_split': [2, 5],
    'randomforestregressor__min_samples_leaf': [1, 2],
    'randomforestregressor__bootstrap': [True, False]
}

# Inicjalizacja RandomizedSearchCV
random_search = RandomizedSearchCV(
    estimator=model_pipeline,
    param_distributions=param_distributions,
    n_iter=10,
    cv=5,
    scoring='neg_mean_squared_error',
    random_state=42,
    n_jobs=-1,
    verbose=1
)

# Trenowanie modelu z optymalizacją hiperparametrów
random_search.fit(X_train, y_train)

# Najlepsze parametry
best_params = random_search.best_params_
logging.info(f'Najlepsze parametry: {best_params}')

# Przewidywanie na zbiorze testowym
y_pred = random_search.predict(X_test)

# Obliczanie metryk
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
mape = mean_absolute_percentage_error(y_test, y_pred)
```

Możliwe przyczyny niezadowalających wyników

- **Ograniczona informatywność cech:**
    - Cechy mogą nie dostarczać wystarczającej ilości informacji do dokładnego przewidywania score.
- **Złożoność zależności:**
    - Relacje między zmiennymi a zmienną docelową mogą być na tyle skomplikowane, że model ich nie uchwycił.
- **Szum w danych:**
    - Obecność szumu może utrudniać modelowi naukę istotnych wzorców.

Alternatywne podejścia

Przetestowano również model **Gradient Boosting** (XGBoost), jednak nie uzyskano znaczącej poprawy:

- **Zbiór testowy (XGBoost):**
    - **MSE:** 48.5271
    - **MAE:** 5.7489
    - **RMSE:** 6.9661
    - **R^2:** 0.3601
    - **MAPE:** 11.91%