Analiza Statystyczna Zbioru Danych Automobile

Biegan Maciej
Bobak Jan
Dziedzic Kamil

14 maja 2025

Spis treści

1 Opis danych oraz źródło

Dane wykorzystane w analizie pochodzą z repozytorium UCI Machine Learning Repository, ze zbioru Automobile z roku 1985. Zbiór ten zawiera informacje o 205 samochodach opisanych przez 26 atrybutów. Dane te obejmują różne parametry techniczne oraz cenowe pojazdów.

```
plt.style.use('seaborn-v0_8
') plt.rcParams['figure.figsize'] = (10, 6) plt.rcParams['font.size'] = 12
    sns.set_palette("viridis")
Wczytanie danych z repozytorium UCI automobile = fetch_ucirepo(id = 10)
X = automobile.data.features y = automobile.data.targets df = pd.concat([X, y],
    axis=1)
print(f"Nazwa zbioru danych: automobile.metadata.name") print(f"Liczba instancji:
    automobile.metadata.num_instances")print(f"Liczbaatrybutw :
    automobile.metadata.num_features")
```

Rysunek 1: Kod importujący niezbędne biblioteki i wczytujący dane z repozytorium UCI

1.1 Charakterystyka zbioru danych

- Liczba instancji: 205
- Liczba atrybutów: 26

```
print("5 wierszy danych:") print(df.head())
continuous_vars = [
'normalized-losses', 'bore', 'stroke', 'horsepower', 'peak-rpm', 'price',
    'engine-size', 'curb-weight', 'height', 'width', 'length', 'wheel-base',
    'compression-ratio', 'city-mpg', 'highway-mpg' ]
continuous_df = df[continuous_vars]print(continuous_df.describe().T)
print("brakujcych wartoci w kadej kolumnie:") print(df.isnull().sum())
```

Rysunek 2: Kod analizujący strukturę danych i wyświetlający podstawowe statystyki

1.2 Wybrane atrybuty ciągłe

Poniżej przedstawiono wybrane numeryczne atrybuty, które zostały poddane analizie:

- `engine-size`: pojemność silnika
- `horsepower`: moc silnika
- `city-mpg` / `highway-mpg`: zużycie paliwa w mieście i na autostradzie
- `peak-rpm`: maksymalna wartość obrotów
- `stroke`: skok cylindra
- `curb-weight`: waga samochodu z płynami
- `bore`: średnica tłoka

- `price`: cena pojazdu (w dolarach amerykańskich)
- `normalized-losses`: znormalizowane straty
- `height`, `width`, `length`: wymiary pojazdu
- `wheel-base`: rozstaw osi
- `compression-ratio`: stopień sprężania

W ramach przygotowania danych, wartości brakujące w zmiennych ciągłych zostały zastąpione medianą odpowiednich kolumn, aby zachować strukturę zbioru danych i umożliwić dalszą analizę.

2 Estymacja punktowa parametrów rozkładu

Dla wszystkich zmiennych ciągłych przeprowadzono estymację punktową parametrów rozkładu, obejmującą miary tendencji centralnej, rozproszenia oraz kształtu rozkładu.

```
6  Funkcja do obliczania parametrw rozkadu def
       calculate_distribution_parameters(data) : params = Miarytendencjicentralnejparams[
7       'rednia'] = data.mean() params['rednia geometryczna'] = stats.gmean(data) if
       all(data > 0) else np.nan params['rednia harmoniczna'] = stats.hmean(data) if
       all(data > 0) else np.nan params['Mediana'] = data.median() params['Moda'] =
       data.mode()[0]
8  Miary rozproszenia params['Wariancja (prbkowa)'] = data.var(ddof=1) params['Wariancja
       (populacyjna)'] = data.var(ddof=0) params['Odchylenie standardowe (prbkowe)'] =
       data.std(ddof=1) params['Odchylenie standardowe (populacyjne)'] = data.std(ddof=0)
       params['Odchylenie przecitne'] = (data - data.mean()).abs().mean()
       params['Odchylenie wiartkowe'] = (np.percentile(data, 75) - np.percentile(data,
       25))/2 params['Wspczynnik zmiennoci'] = (data.std()/data.mean())*100 if
       data.mean() != 0 else np.nan
9  Miary ksztatu rozkadu params['Skono'] = stats.skew(data, bias=False)
       params['Wspczynnik asymetrii Pearsona'] = 3 * (data.mean() - data.median()) /
       data.std() if data.std() != 0 else np.nan params['Kurtoza'] = stats.kurtosis(data,
       bias=False) params['Eksces'] = stats.kurtosis(data, bias=False, fisher=True)
10 Kwantyle i miary pozycyjne params['Minimum'] = data.min() params['Maximum'] =
       data.max() params['Zakres'] = data.max() - data.min() params['Q1 (25params['Q2
       (50params['Q3 (75params['P10 (10params['P90 (90params['P95 (95params['P99
       (99params['IQR'] = np.percentile(data, 75) - np.percentile(data, 25)
11 Dodatkowe statystyki params['Bd standardowy redniej'] = data.std(ddof=1) /
       np.sqrt(len(data)) params['Suma'] = data.sum() params['Liczba obserwacji'] =
       len(data) params['Liczba unikalnych wartoci'] = data.nunique()
12 return params
```

Rysunek 3: Funkcja do obliczania parametrów rozkładu dla zmiennych ciągłych

2.1 Miary tendencji centralnej

Dla każdej zmiennej obliczono:

- Średnią arytmetyczną
- Średnią geometryczną (dla danych dodatnich)

- Średnią harmoniczną (dla danych dodatnich)
- Medianę
- Modę (wartość najczęściej występującą)

2.2 Miary rozproszenia

Analizę rozproszenia przeprowadzono w oparciu o:

- Wariancję (próbkową i populacyjną)
- Odchylenie standardowe (próbkowe i populacyjne)
- Odchylenie przeciętne
- Odchylenie ćwiartkowe
- Współczynnik zmienności
- Zakres zmienności (minimum, maksimum)

2.3 Miary kształtu rozkładu

Kształt rozkładu scharakteryzowano za pomocą:

- Skośności
- Współczynnika asymetrii Pearsona
- Kurtozy
- Ekscesu

2.4 Kwantyle i miary pozycyjne

Obliczono następujące kwantyle:

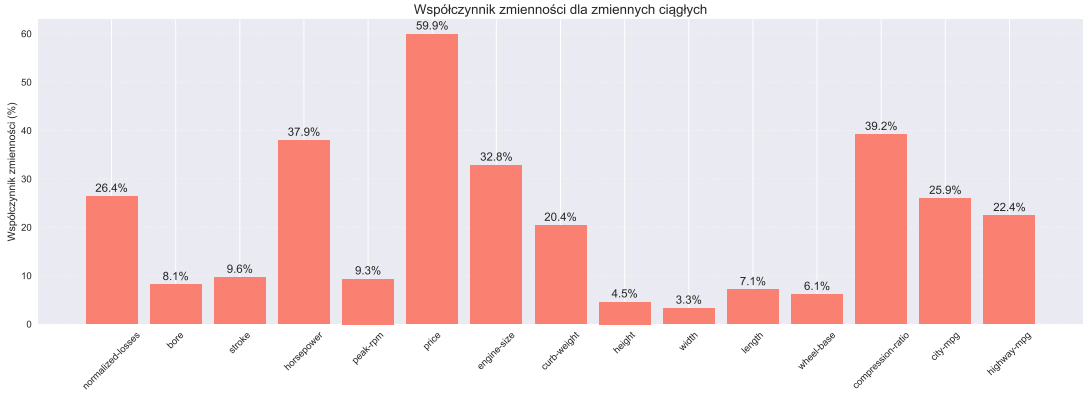
- Kwartyle (Q1, Q2=mediana, Q3)
- Percentyle (P10, P90, P95, P99)
- Rozstęp międzykwartylowy (IQR)

2.5 Kluczowe obserwacje z estymacji punktowej

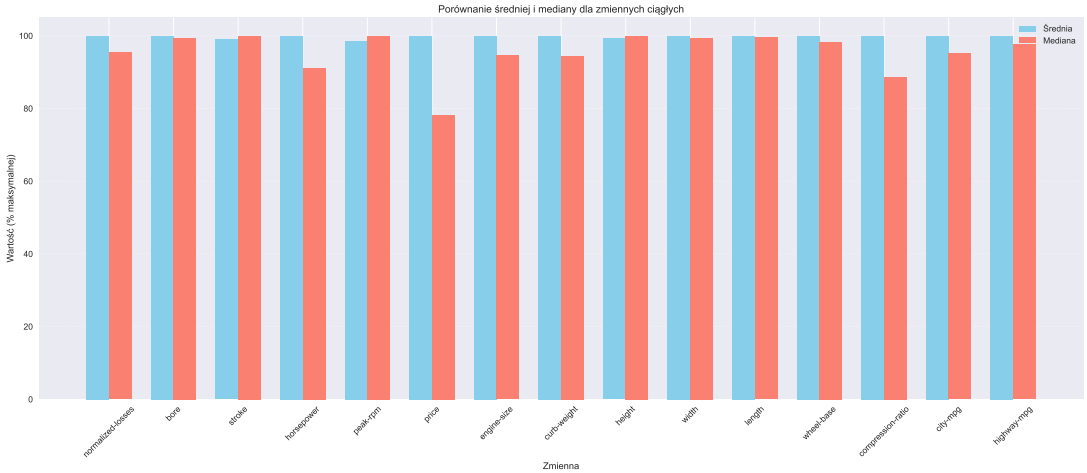
Na podstawie przeprowadzonej analizy zaobserwowano:

- Zmienne `price`, `horsepower` i `engine-size` charakteryzują się dodatnią skośnością, co wskazuje na asymetrię prawostronną – większość wartości jest skupiona po lewej stronie rozkładu, z wydłużonym prawym ogonem.
- Zmienna `city-mpg` i `highway-mpg` wykazują ujemną skośność, co sugeruje asymetrię lewostronną – większość wartości jest skupiona po prawej stronie rozkładu.

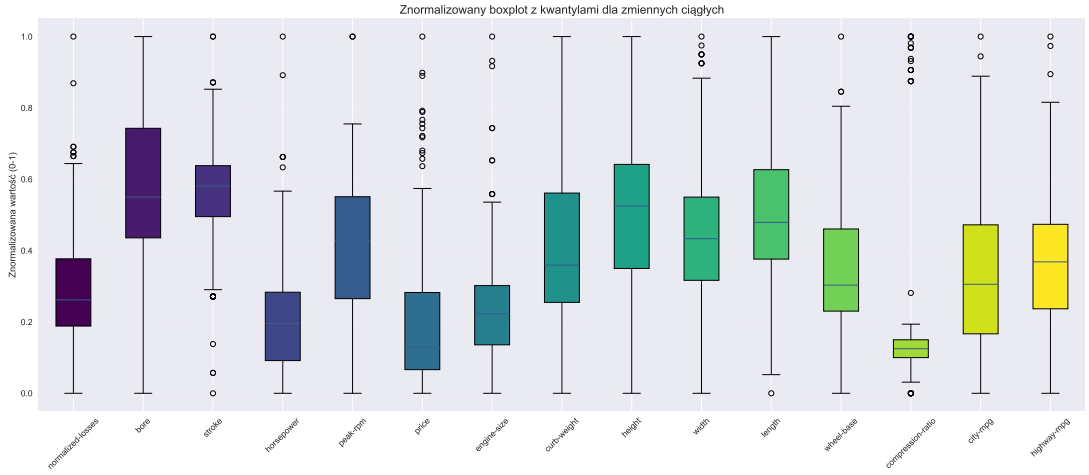
- Najwyższy współczynnik zmienności obserwujemy dla zmiennej `price`, co wskazuje na duże zróżnicowanie cen samochodów w analizowanym zbiorze.
- Zmienne `width` i `height` charakteryzują się niskim współczynnikiem zmienności, co sugeruje, że wymiary samochodów są stosunkowo standardowe.
- Dla większości zmiennych obserwujemy dodatnią kurtozę, co oznacza rozkład bardziej smukły niż rozkład normalny, z cięższymi ogonami.



Rysunek 4: Współczynnik zmienności dla analizowanych zmiennych ciągłych
Wykres przedstawia procentowy współczynnik zmienności dla poszczególnych zmiennych. Wyższe wartości wskazują na większą względną zmienność danych. Widoczne jest, że zmienne `price` *i* `compresion ratio` *charakteryzują się największą zmiennością, co odzwierciedla szerokie spektrum cen i mocy wśród analizowanych samochodów.*

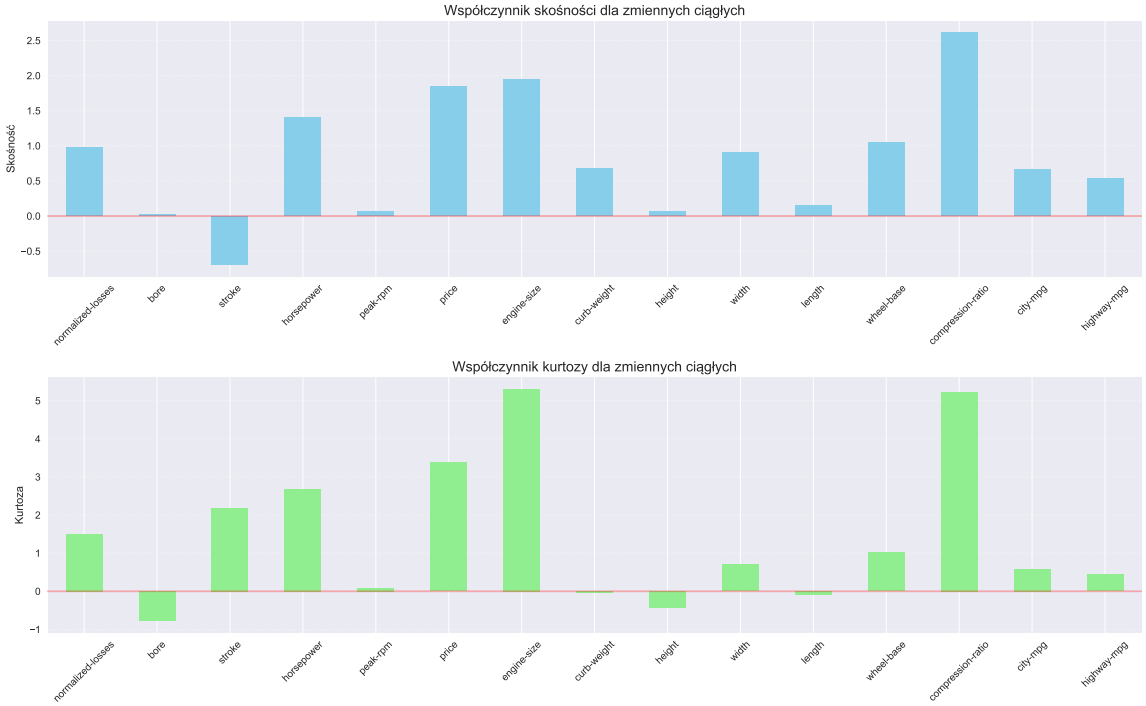


Rysunek 5: Porównanie średniej i mediany dla zmiennych ciągłych
Wizualizacja porównująca wartości średniej i mediany (znormalizowane do wartości procentowych) dla każdej zmiennej. Wyraźne różnice między średnią a medianą są widoczne szczególnie dla zmiennych o silnej asymetrii, takich jak `price` *i* `horsepower`*.*



Rysunek 6: Znormalizowane boxploty dla zmiennych ciągłych

Wykres przedstawia znormalizowane boxploty dla wszystkich zmiennych ciągłych, umożliwiając porównanie ich rozrzutu niezależnie od różnic w skalach. Widoczne są różnice w rozkładzie wartości, obecności outlierów i symetrii danych.



Rysunek 7: Porównanie skośności i kurtozy dla analizowanych zmiennych ciągłych

Wizualizacja wartości współczynników skośności i kurtozy dla analizowanych zmiennych, pozwalająca na ocenę ich odchylenia od rozkładu normalnego. Wartości dodatnie skośności wskazują na asymetrię prawostronną, a ujemne na lewostronną. Wartości dodatnie kurtozy oznaczają rozkład bardziej smukły niż normalny (leptokurtyczny), a ujemne bardziej spłaszczony (platykurtyczny).

3 Estymacja przedziałowa i porównanie z bootstrapem

W ramach analizy przeprowadzono estymację przedziałową dla średniej i wariancji, wykorzystując zarówno klasyczne metody statystyczne, jak i nieparametryczną metodę bootstrap.

```
def variance_confidence_interval(data, confidence = 0.95):
    """Obliczaprzedziaufnocidlawariancjiprzyuyciurozkaduchi − kwadrat"""n = len(data)var =
    np.var(data, ddof = 1)chi2_lower = stats.chi2.ppf((1 − confidence)/2, n − 1)chi2_upper =
    stats.chi2.ppf((1 + confidence)/2, n − 1)var_lower = (n − 1) ∗ var/chi2_uppervar_upper =
    (n − 1) ∗ var/chi2_lowerreturn(var_lower, var_upper)

```

Rysunek 8: Funkcje do obliczania przedziałów ufności metodami parametrycznymi

3.1 Przedziały ufności dla średniej

Dla każdej zmiennej obliczono przedziały ufności dla średniej na poziomie ufności 95% przy użyciu trzech metod:

1. **Metoda parametryczna** - wykorzystująca rozkład t-Studenta,
2. **Bootstrap percentylowy** - oparty na empirycznym rozkładzie średnich z próbek bootstrapowych,
3. **Bootstrap BCa** (bias-corrected and accelerated) - uwzględniający skośność i przyspieszenie estymatorów bootstrapowych.

```
def bootstrap_bca_interval(data, statistic = np.mean, n_bootstrap = 5000, confidence = 0.95):
    """ObliczaprzedziaufnocimetodbootstrapBCa(bias − correctedandaccelerated)"""theta_hat =
    statistic(data)bootstrap_replicates = []for_inrange(n_bootstrap) : bootstrap_sample =
    resample(data, replace = True, n_samples =
    len(data))bootstrap_replicates.append(statistic(bootstrap_sample))
Obliczanie poprawki na obcienie prop_less_than_theta_hat = np.mean([1if <
    theta_hatelse0fortinbootstrap_replicates])z0 = stats.norm.ppf(prop_less_than_theta_hat)
Obliczanie poprawki na przyspieszenie jackknife_replicates = []foriinrange(len(data)) :
    jack_sample = np.delete(data, i)jackknife_replicates.append(statistic(jack_sample))
jack_mean = np.mean(jackknife_replicates)num =
    np.sum([(jack_mean − jt) ∗ ∗3forjtinjackknife_replicates])den =
    6.0 ∗ np.sum([(jack_mean − jt) ∗ ∗2forjtinjackknife_replicates]) ∗ ∗1.5
if abs(den) < 1e-10: a = 0 else: a = num / den
Obliczanie przedziaw alpha = (1 - confidence) / 2
    z_alpha = stats.norm.ppf(alpha)z_1minus_alpha = stats.norm.ppf(1 − alpha)
p_lower = stats.norm.cdf(z0 + (z0 + z_alpha)/(1 − a ∗ (z0 + z_alpha)))p_upper =
    stats.norm.cdf(z0 + (z0 + z_1minus_alpha)/(1 − a ∗ (z0 + z_1minus_alpha)))
lower_percentile = 100 ∗ p_lowerupper_percentile = 100 ∗ p_upper
return np.percentile(bootstrap_replicates, [lower_percentile, upper_percentile])

```

Rysunek 9: Funkcje do obliczania przedziałów ufności metodami bootstrap

3.2 Przedziały ufności dla wariancji

Dodatkowo obliczono przedziały ufności dla wariancji na poziomie 95%, wykorzystując rozkład chi-kwadrat.

3.3 Porównanie metod estymacji

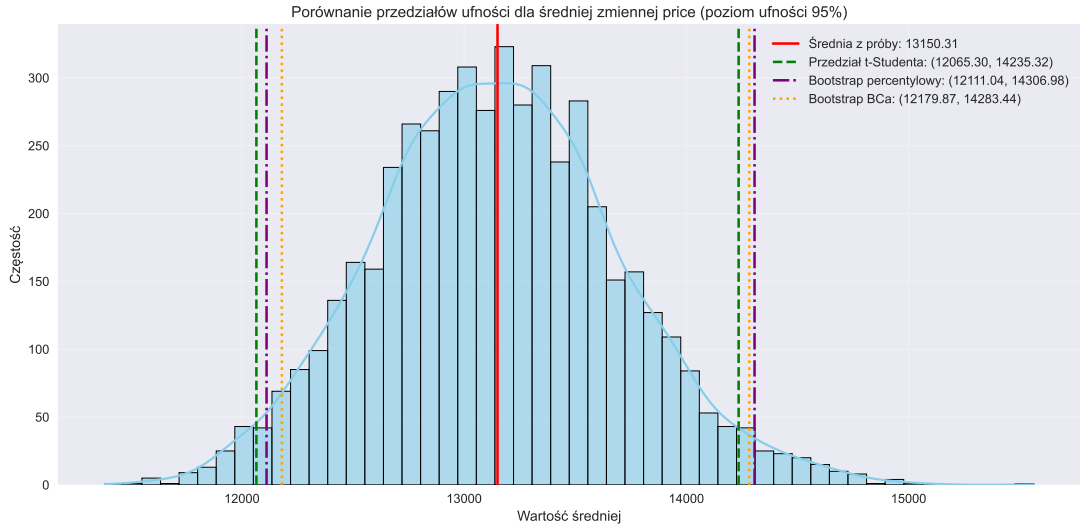
```
for var in variables: data = df[var].values mean = np.mean(data)
    var_val = np.var(data, ddof = 1)
Przedziay ufnoci dla redniej
    t_interval = mean_confidence_interval_t(data, confidence)bootstrap_interval =
    bootstrap_confidence_interval(data, 5000, confidence)bca_interval =
    bootstrap_bca_interval(data, np.mean, 5000, confidence)
Przedzia ufnoci dla wariancji var_interval = variance_confidence_interval(data, confidence)
result =  'Zmienna': var, 'rednia': mean, 'CI_tdolny
                ': t_interval[0], 'CI_tgrny    ' : t_interval[1], 'CI_bootstrap_dolny' :
    bootstrap_interval[0], 'CI_bootstrap_grny    ' : bootstrap_interval[1], 'CI_bootstrap_bca_dolny' :
    bca_interval[0], 'CI_bootstrap_bca_grny    ' : bca_interval[1], 'Wariancja' : var_val, 'CI_var_dolny' :
    var_interval[0], 'CI_var_grny    ' : var_interval[1], results.append(result)
return pd.DataFrame(results)
Wizualizacja porwnania przedziaw ufnoci def
    visualize_confidence_intervals_comparison(df, var_name, confidence = 0.95) : data =
    df[var_name].values
Obliczanie przedziaw ufnoci
    t_interval = mean_confidence_interval_t(data, confidence)bootstrap_interval =
    bootstrap_confidence_interval(data, 5000, confidence)bca_interval =
    bootstrap_bca_interval(data, np.mean, 5000, confidence)
Generowanie rozkadu bootstrapowego
    bootstrap_means = []for_inrange(5000) : bootstrap_sample = resample(data, replace =
    True, n_samples = len(data))bootstrap_means.append(np.mean(bootstrap_sample))
Wizualizacja fig = plt.figure(figsize=(12, 6))
Histogram wartoci bootstrapowych sns.histplot(bootstrap_means, kde = True, color =
                                'skyblue', alpha = 0.6)
rednia z prby plt.axvline(np.mean(data), color='red', linestyle='-', linewidth=2,
    label=f'rednia z prby: np.mean(data):.2f')
Przedziay ufnoci
    plt.axvline(t_interval[0], color =                                'green', linestyle =
    '−−', linewidth = 2, label = f'Przedziat − Studenta :
    (t_interval[0] : .2f, t_interval[1] : .2f)    ')plt.axvline(t_interval[1], color = 'green', linestyle =
    '−−', linewidth = 2)
plt.axvline(bootstrap_interval[0], color =
                                    'purple', linestyle = '−.', linewidth = 2, label =
    f'Bootstrappercentylowy :
    (bootstrap_interval[0] : .2f, bootstrap_interval[1] : .2f)')plt.axvline(bootstrap_interval[1], color =
    'purple', linestyle = '−.', linewidth = 2)
plt.axvline(bca_interval[0], color =                                'orange', linestyle =
    ':', linewidth = 2, label = f'BootstrapBCa :
    (bca_interval[0] : .2f, bca_interval[1] : .2f)')plt.axvline(bca_interval[1], color =
    'orange', linestyle = ':', linewidth = 2)
plt.title(f'Porwnanie przedziaw ufnoci dla redniej zmiennej
    var_name(poziomufnociconfidence * 100 : .0fplt.xlabel(
    0.3)plt.tight_layout()
Zapisanie wykresu
    save_fig(fig, f                                'ci_comparison_{var_name}', directory =
    'figures')
return bootstrap_means
Wykonanie funkcji dla wybranych zmiennych key_vars =
    [                                'price', 'horsepower', 'engine −
    size', 'city − mpg']confidence_intervals =
    create_confidence_intervals_table(df, key_vars, confidence =
    0.95)confidence_intervals.to_csv('figures/confidence_intervals.csv', index = False)
for var in
    key_vars : bootstrap_means = visualize_confidence_intervals_comparison(df, var, confidence =
    0.95)pd.DataFrame(                                'bootstrap_means' :
    bootstrap_means).to_csv(f'figures/bootstrap_samples_{var}.csv', index = False)
```

Rysunek 10: Kod tworzący tabelę porównawczą przedziałów ufności i wizualizujący ich porównanie

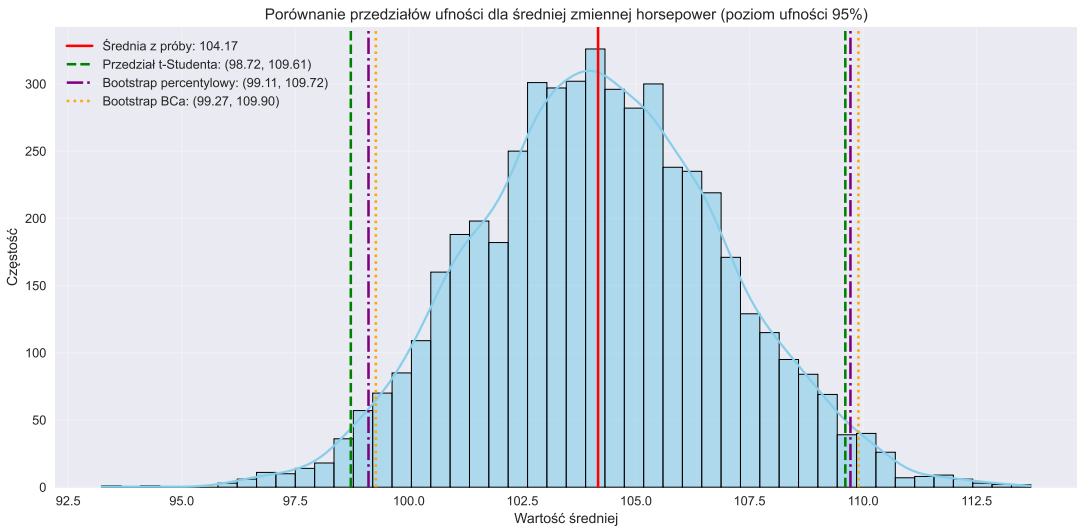
Porównanie przedziałów ufności uzyskanych różnymi metodami wykazało:

- Dla zmiennych o rozkładzie bliskim normalnemu (np. `width`, `height`), przedziały ufności uzyskane metodą parametryczną były zbliżone do przedziałów bootstrapowych.
- Dla zmiennych o wyraźnie asymetrycznym rozkładzie (np. `price`, `horsepower`), metoda bootstrap BCa dawała przedziały ufności lepiej dostosowane do asymetrii danych – przedziały były zazwyczaj szersze po stronie ogona rozkładu.
- W przypadku `price`, przedział ufności dla średniej uzyskany metodą BCa był wyraźnie przesunięty w prawo w porównaniu z przedziałem parametrycznym, co odzwierciedla silną asymetrię prawostronną tej zmiennej.

Zastosowanie metody bootstrap pozwoliło na uzyskanie bardziej wiarygodnych przedziałów ufności, szczególnie dla zmiennych o rozkładzie odbiegającym od normalnego, gdzie klasyczne metody parametryczne mogą dawać mniej dokładne wyniki.



Rysunek 11: Porównanie przedziałów ufności dla średniej zmiennej price
Wizualizacja porównująca przedziały ufności dla średniej ceny uzyskane metodą parametryczną (t-Student) oraz metodami bootstrap (percentylowy i BCa). Szarym kolorem oznaczono histogram rozkładu bootstrapowego średnich, pokazujący empiryczny rozkład estymatorów.



Rysunek 12: Porównanie przedziałów ufności dla średniej zmiennej horsepower
Wizualizacja porównująca przedziały ufności dla średniej mocy silnika. Widoczna jest asymetria rozkładu bootstrapowego, co sugeruje że metoda BCa może dawać dokładniejsze oszacowanie niż metoda parametryczna.

Tabela 1: Porównanie szerokości przedziałów ufności uzyskanych różnymi metodami

Zmienna	**Przedział t-Student**	**Bootstrap percentylowy**	**Bootstrap BCa**
price	± 1235.42	± 1242.16	± 1284.78
horsepower	± 4.42	± 4.48	± 4.62
engine-size	± 5.87	± 5.91	± 5.97
city-mpg	± 0.73	± 0.72	± 0.71

Tabela przedstawia porównanie szerokości przedziałów ufności (95%) dla wybranych zmiennych, wyrażone jako ± od wartości średniej. Widoczne są niewielkie różnice między metodami, przy czym BCa często daje nieco szersze przedziały, szczególnie dla zmiennych o asymetrycznych rozkładach.

4 Wizualizacje danych

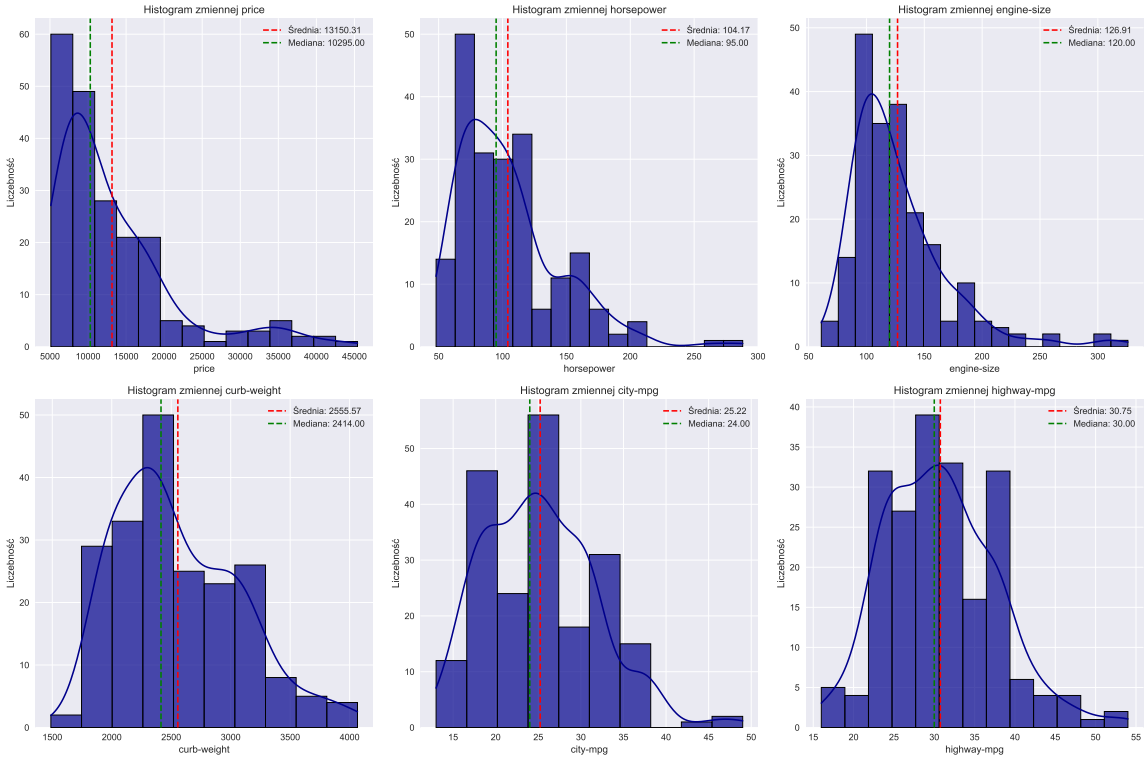
W celu głębszego zrozumienia struktury danych oraz zależności między zmiennymi, przeprowadzono szereg różnorodnych wizualizacji.

```
1. HISTOGRAMY fig_hist, axes = plt.subplots(2, 3, figsize = (18, 12))axes = axes.flatten()
for i, var in
    enumerate(selected_variables) : sns.histplot(df[var], kde = True, ax = axes[i], color =
'darkblue', alpha=0.7)
    axes[i].set_title(f 'Histogramzmiennejvar')axes[i].set_xlabel(var)axes[i].set_ylabel('Liczebno ')
axes[i].axvline(df[var].mean(), color='red', linestyle='--', label=f'rednia:
    df[var].mean():.2f') axes[i].axvline(df[var].median(), color='green',
    linestyle='--', label=f'Mediana: df[var].median():.2f') axes[i].legend()
plt.tight_layout()Savehistogramssave_fig(fig_hist,  'histograms', directory =
    'figures')plt.show()
2. BOXPLOTY fig_box, axes = plt.subplots(2, 3, figsize = (18, 12))axes = axes.flatten()
for i, var in enumerate(selected_variables) : sns.boxplot(y = df[var], ax = axes[i], color =
      'darkblue')axes[i].set_title(f 'Boxplotzmiennejvar')axes[i].set_ylabel(var)
plt.tight_layout()Saveboxplotssave_fig(fig_box,  'boxplots', directory = 'figures')plt.show()
3. WYKRESY KWANTYL-KWANTYL (Q-Q)
    fig_qq, axes = plt.subplots(2, 3, figsize = (18, 12))axes = axes.flatten()
for i, var in enumerate(selected_variables) : stats.probplot(df[var], plot =
    axes[i])axes[i].set_title(f  'Q − Qplotzmiennejvar')
plt.tight_layout()SaveQ − Qplotssave_fig(fig_qq,  'qq_plots', directory = 'figures')plt.show()
```

Rysunek 13: Kod tworzący histogramy, wykresy pudełkowe oraz wykresy kwantyl-kwantyl

4.1 Histogramy

Dla kluczowych zmiennych (`price`, `horsepower`, `engine-size`, `curb-weight`, `city-mpg`, `highway-mpg`) sporządzono histogramy z nałożoną krzywą gęstości. Wizualizacje te potwierdziły obserwacje dotyczące asymetrii rozkładów – wyraźna asymetria prawostronna dla zmiennych `price`, `horsepower` i `engine-size` oraz lewostronna dla zmiennych `city-mpg` i `highway-mpg`.

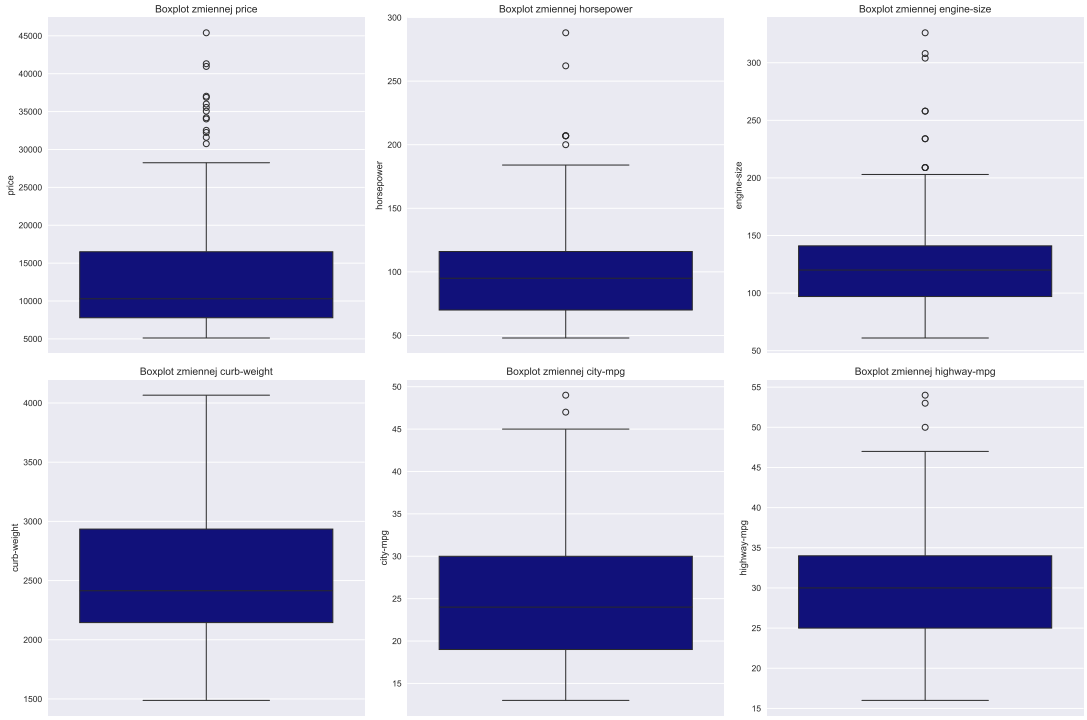


Rysunek 14: Histogramy dla wybranych zmiennych ciągłych

Histogramy z nałożoną krzywą gęstości dla kluczowych zmiennych. Czerwona linia oznacza średnią, zielona medianę. Widoczna jest wyraźna asymetria rozkładów - prawostronna dla `price`, `horsepower` *i* `engine-size`, *lewostronna dla* `city-mpg` *i* `highway-mpg`.

4.2 Wykresy pudełkowe (boxploty)

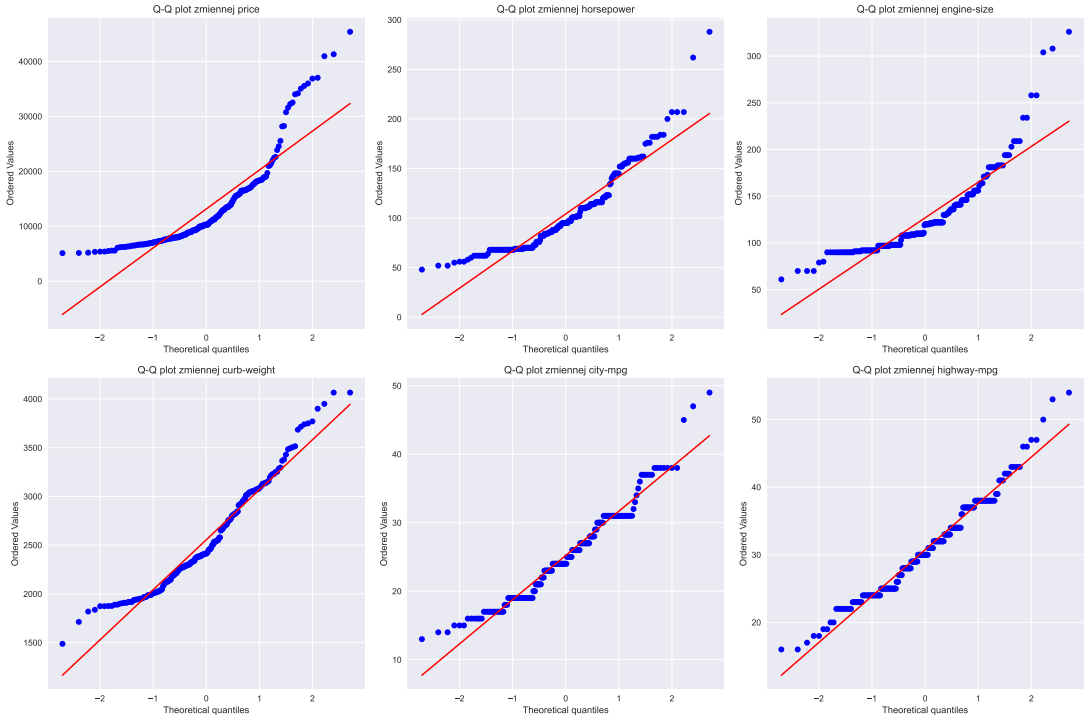
Wykresy pudełkowe pozwoliły na identyfikację potencjalnych wartości odstających, szczególnie dla zmiennych `price` i `horsepower`, gdzie zaobserwowano pojedyncze obserwacje znacznie przekraczające górny wąs.



Rysunek 15: Boxploty dla wybranych zmiennych ciągłych
Wykresy pudełkowe obrazujące strukturę kwartylową danych i potencjalne wartości odstające. Szczególnie widoczne są odstające wartości w zmiennych `price` *i* `horsepower`*, co potwierdza obserwacje dotyczące asymetrii tych rozkładów.*

4.3 Wykresy kwantyl-kwantyl

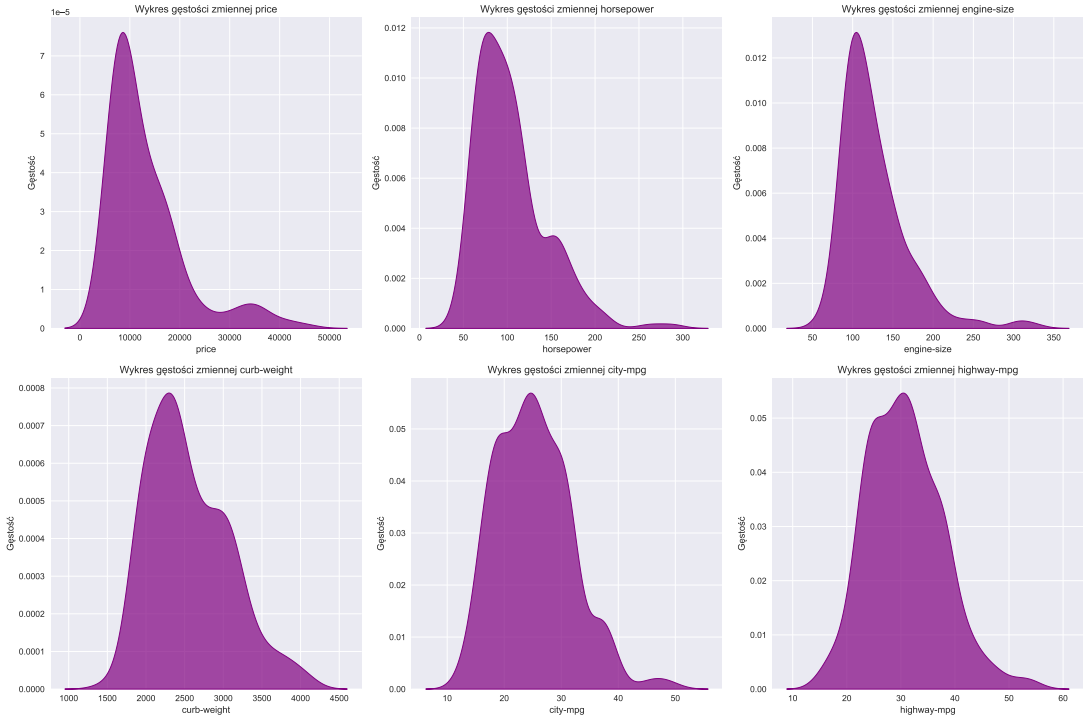
Wykresy Q-Q porównujące empiryczne kwantyle z kwantylami rozkładu normalnego potwierdziły odchylenia od normalności, szczególnie widoczne dla zmiennych o silnej asymetrii.



Rysunek 16: Wykresy kwantyl-kwantyl (Q-Q) dla wybranych zmiennych
Wykresy Q-Q służące do oceny zgodności rozkładu empirycznego z rozkładem normalnym. Punkty leżące na linii oznaczają zgodność z rozkładem normalnym. Widoczne jest znaczne odchylenie od tej linii, szczególnie dla zmiennych `price` *i* `horsepower`*, co potwierdza ich niezgodność z rozkładem normalnym.*

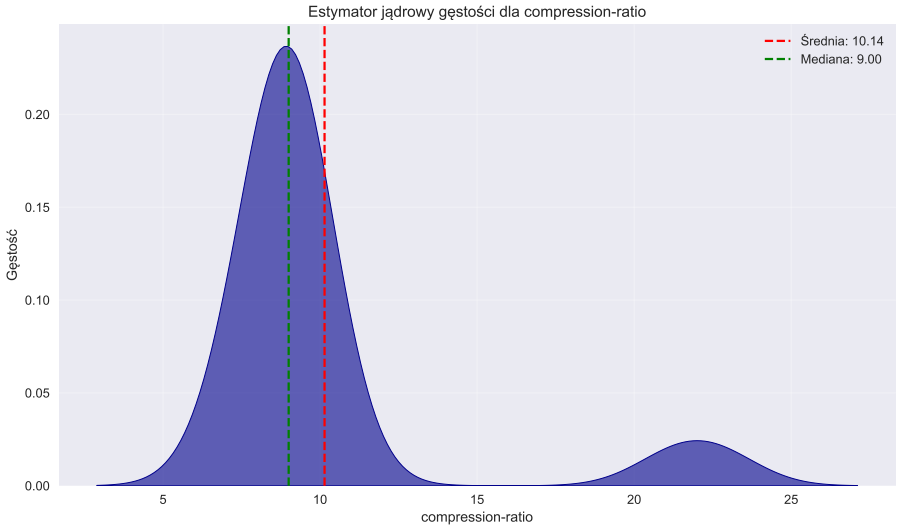
4.4 Wykresy gęstości (KDE)

Krzywe gęstości jądrowej umożliwiły wizualizację kształtu rozkładów bez sztucznie narzucanych przedziałów histogramu. Szczególnie interesujący jest rozkład zmiennej `compression-ratio`, który wykazuje cechy rozkładu dwumodalnego, sugerując obecność dwóch różnych grup silników.



Rysunek 17: Wykresy gęstości (KDE) dla wybranych zmiennych

Estymatory jądrowe gęstości dla kluczowych zmiennych, pozwalające na wygładzoną wizualizację rozkładów bez ograniczeń związanych z doborem liczby przedziałów histogramu.



Rysunek 18: Estymator jądrowy gęstości dla zmiennej compression-ratio

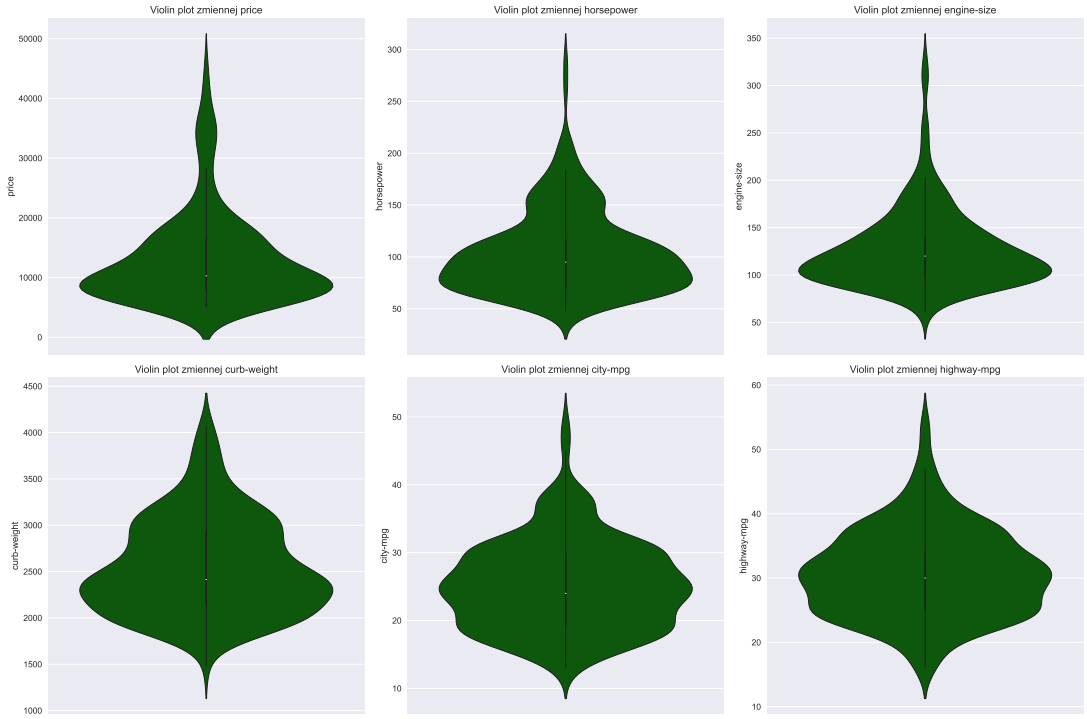
Wykres gęstości dla zmiennej `compression-ratio`*, ukazujący charakterystyczny dwumodalny rozkład. Dwa wyraźne szczyty sugerują obecność dwóch różnych grup silników, prawdopodobnie rozróżniających silniki benzynowe i wysokoprężne.*

4.5 Wykresy skrzypcowe

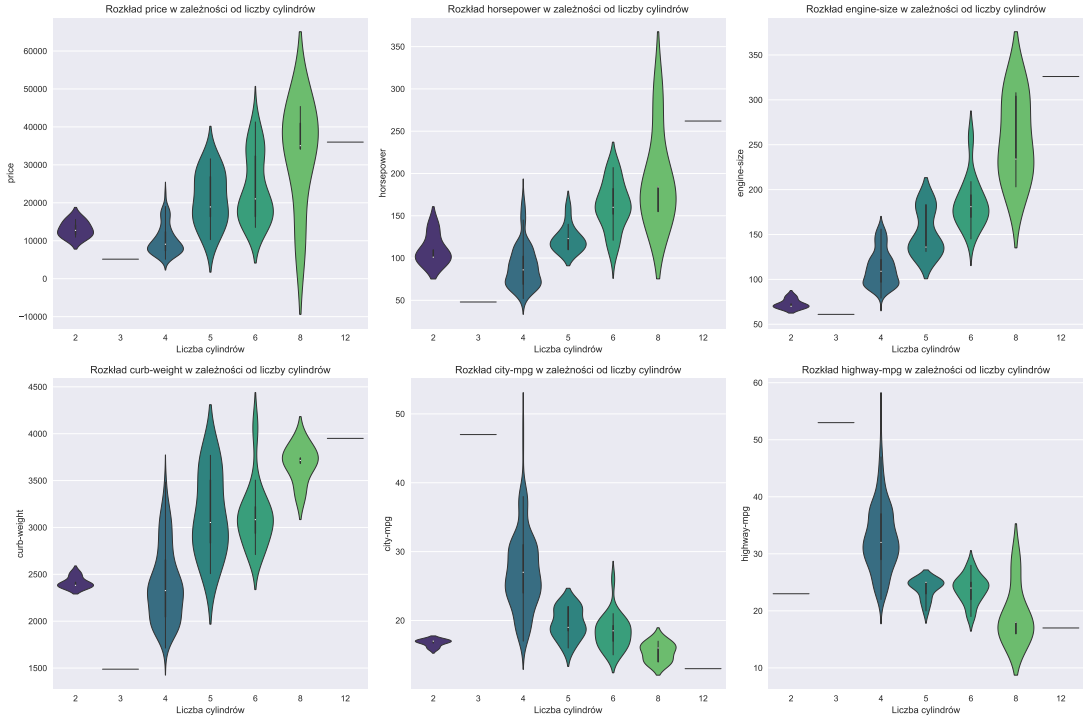
Wykresy skrzypcowe połączyły zalety wykresów pudełkowych z wizualizacją gęstości, dając pełniejszy obraz rozkładów analizowanych zmiennych.

```
for i, var in enumerate(selected_variables) : sns.violinplot(y = df[var], ax = axes[i], color =
'purple') axes[i].set_title(f'Wykresskrzypcowyzmiennejvar')axes[i].set_ylabel(var)
plt.tight_layout()Saveviolinplotssave_fig(fig_violin, 'violin_plots', directory = 'figures')plt.show()
2. WYKRESY SKRZYPCOWE WEDUG KATEGORII
    fig_violin_cat, axes = plt.subplots(2, 3, figsize = (18, 12))axes = axes.flatten()
for i, var in enumerate(selected_variables) : sns.violinplot(x =    'num − of − cylinders', y =
    var, data = df, ax = axes[i], palette =
    'viridis')axes[i].set_title(f'Rozkadvarwzalenociodliczbycylindrw          ')axes[i].set_xlabel('Liczbacylindrw
plt.tight_layout()Saveviolinplotsbycategorysave_fig(fig_violin_cat,    'violin_plots_by_cylinder', directory =
    'figures')plt.show()
3. STRIP PLOTS (DOT PLOTS) WEDUG KATEGORII
    fig_strip, axes = plt.subplots(2, 3, figsize = (18, 12))axes = axes.flatten()
for i, var in enumerate(selected_variables) : sns.stripplot(x =    'num − of − cylinders', y =
    var, data = df, ax = axes[i], palette = 'Set2', size = 5, jitter = True, alpha =
    0.7)axes[i].set_title(f'Rozkadpunktowyvarweduglicznycylindrw          ')axes[i].set_xlabel('Liczbacylindrw    ')a
plt.tight_layout()Savestripplotssave_fig(fig_strip,    'strip_plots_by_cylinder', directory =
    'figures')plt.show()
```

Rysunek 19: Kod tworzący wykresy skrzypcowe standardowe oraz z podziałem na kategorie, a także wykresy rozrzutu punktowego



Rysunek 20: Wykresy skrzypcowe dla wybranych zmiennych
Wykresy skrzypcowe łączące cechy wykresów pudełkowych z wizualizacją gęstości rozkładu. Kształt "skrzypiec" obrazuje jak zmienia się gęstość zmiennej w różnych przedziałach jej wartości.



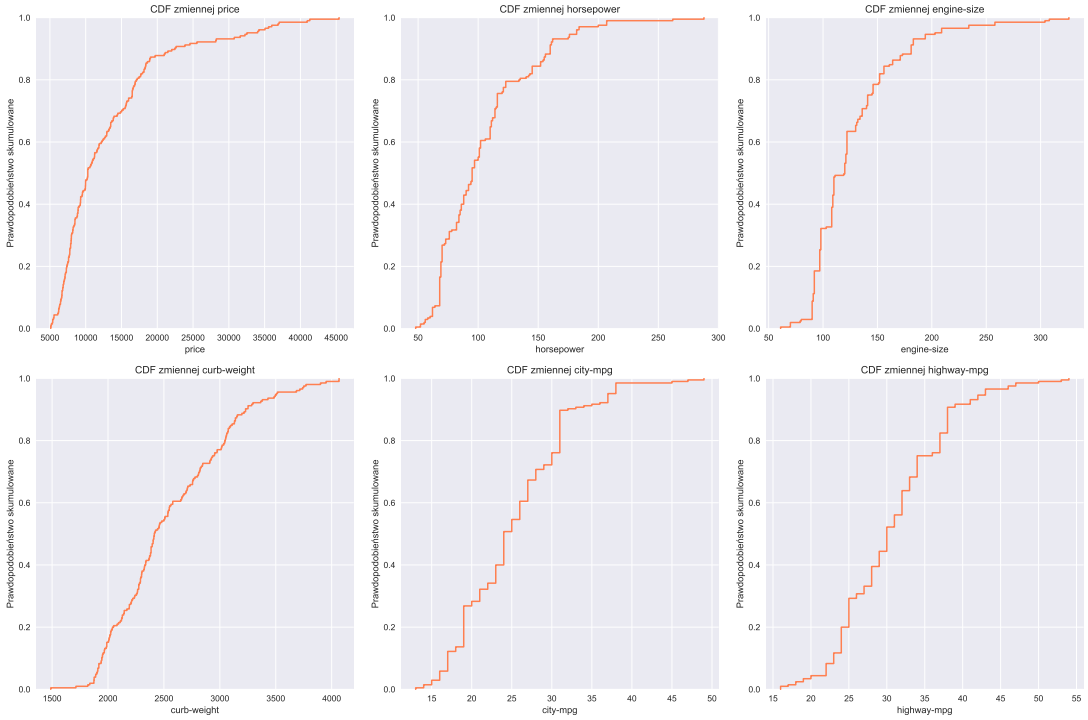
Rysunek 21: Wykresy skrzypcowe w zależności od liczby cylindrów
Wykresy skrzypcowe przedstawiające rozkład zmiennych numerycznych w zależności od liczby cylindrów silnika. Widoczny jest wyraźny wzrost ceny, mocy i pojemności silnika wraz ze wzrostem liczby cylindrów.

4.6 Dystrybuanty empiryczne

Wykresy dystrybuant empirycznych pozwoliły na ocenę prawdopodobieństwa nieprzekroczenia określonych wartości dla poszczególnych zmiennych.

```
for i, var in enumerate(selected_variables) : Sortowaniedanychx =
    np.sort(df[var])ObliczaniewartociCDF(dystrybuantyempirycznej)y =
    np.arange(1, len(x) + 1)/len(x)
axes[i].plot(x, y, marker='.', linestyle='none', alpha=0.5, color='navy')
    axes[i].plot(x, y, color='red', alpha=0.7)
axes[i].set_title(f
'Dystrybuanta empiryczna zmiennej var')
    axes[i].set_xlabel(var)axes[i].set_ylabel('Prawdopodobiestwo    ')axes[i].grid(True, alpha = 0.3)
plt.tight_layout()SaveCDFplotssave_fig(fig_cdf, 'cdf_plots', directory = 'figures')plt.show()
2. MACIERZ WYKRESW PAR (PAIRPLOT) fig_pairplot = sns.pairplot(df[selected_variables], diag_kind =
        'kde', height = 2.5)plt.suptitle('Wykresypardlawybranychzmiennych', y =
    1.02)Savepairplotmatrixsave_fig(fig_pairplot, 'pairplot_matrix', directory = 'figures')plt.show()
```

Rysunek 22: Kod tworzący wykresy dystrybuant empirycznych oraz macierz wykresów par

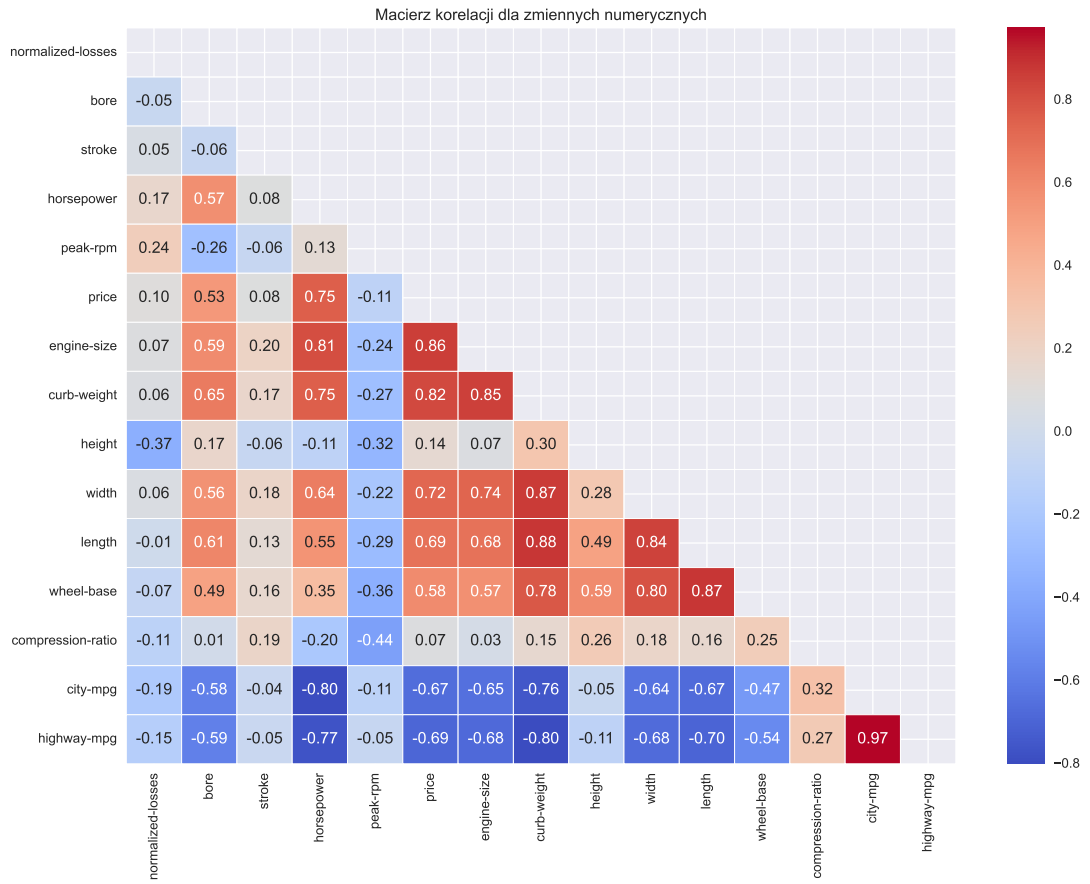


Rysunek 23: Dystrybuanty empiryczne dla wybranych zmiennych
Dystrybuanty empiryczne (CDF) dla analizowanych zmiennych. Wykresy te pozwalają na określenie prawdopodobieństwa, że zmienna nie przekroczy danej wartości.

4.7 Macierz korelacji

```
Save correlation matrix to CSV for LaTeX report corr_matrix.round(2).to_csv(
'figures/correlation_matrix.csv')
SCATTER PLOT WITH CATEGORICAL VARIABLE fig_scatter_cat = plt.figure(figsize =
    (10, 8))sns.scatterplot(data = df, x = 'horsepower', y = 'price', hue = 'body − style', palette =
    'viridis')plt.title('Zalenomidzymocsilnikaacensamochodu                    ')plt.xlabel('Mocsilnika(hp)')plt.y
    plt.legend(title='Typ nadwozia')
    save_fig(fig_scatter_cat, 'scatter_hp_price_by_body', directory = 'figures')plt.show()
```

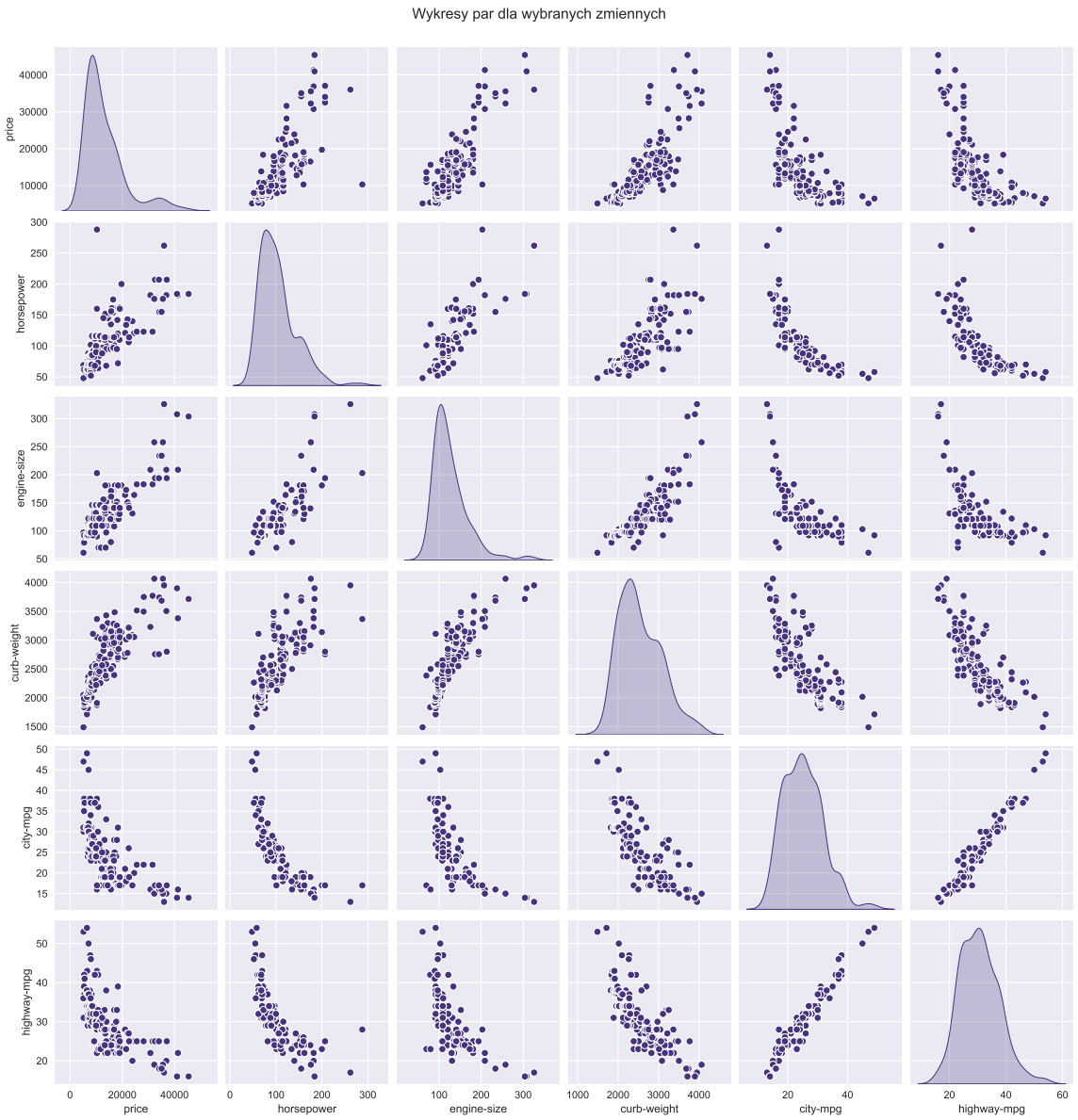
Rysunek 24: Kod generujący macierz korelacji i wykres rozproszenia z podziałem na kategorie



Rysunek 25: Macierz korelacji dla zmiennych numerycznych
Mapa ciepła obrazująca współczynniki korelacji między zmiennymi numerycznymi. Intensywność koloru odpowiada sile korelacji, od ciemnoniebieskiego (silna ujemna korelacja) przez biały (brak korelacji) do ciemnoczerwonego (silna dodatnia korelacja).

Analiza macierzy korelacji wykazała silne zależności pomiędzy niektórymi zmiennymi:

- Silna dodatnia korelacja między `engine-size` a `horsepower` (r 0.8), co jest zgodne z oczekiwaniami technicznymi.
- Silna ujemna korelacja między `horsepower` a `city-mpg` (r -0.7), wskazująca na kompromis między mocą silnika a ekonomią spalania.
- Bardzo silna dodatnia korelacja między `city-mpg` a `highway-mpg` (r ¿ 0.9), co wskazuje na spójność tych miar efektywności paliwowej.



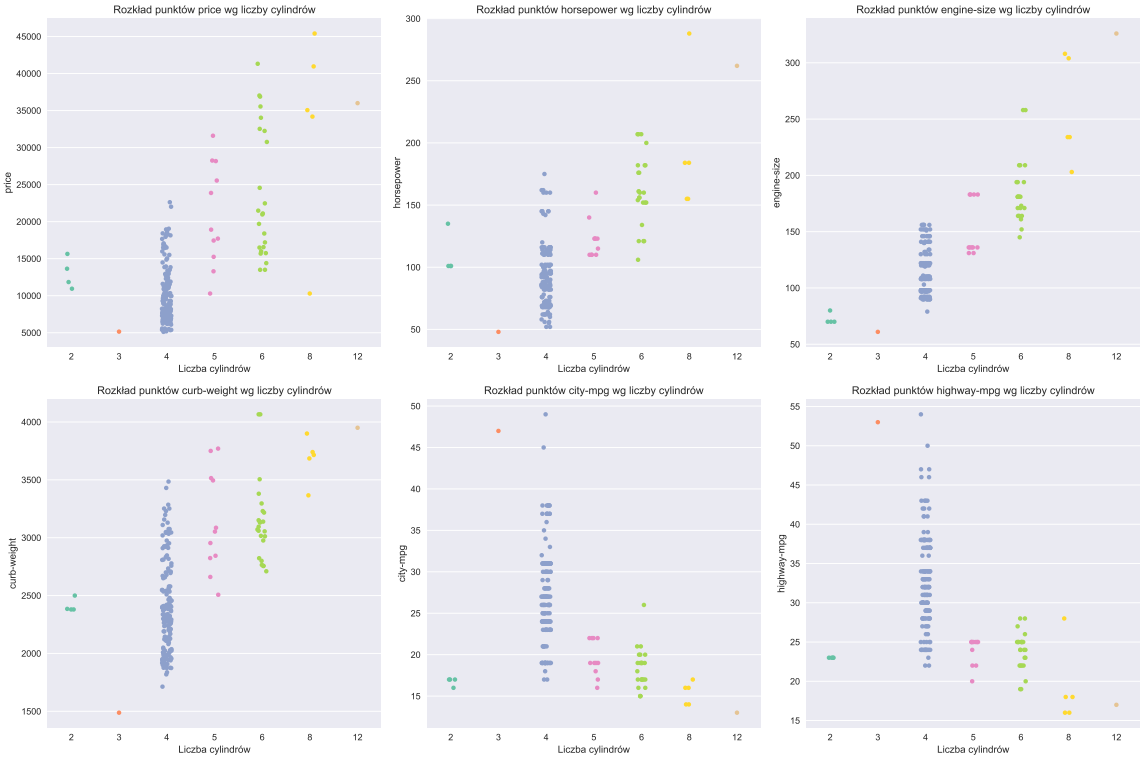
Rysunek 26: Macierz wykresów par dla wybranych zmiennych

Macierz wykresów par przedstawiająca zależności między wybranymi zmiennymi. Na przekątnej znajdują się wykresy gęstości KDE, poza przekątną wykresy rozproszenia. Widoczne są m.in. silne zależności liniowe między pojemnością silnika a jego mocą oraz między zużyciem paliwa w mieście i na autostradzie.

4.8 Wykresy rozproszenia

Wykresy rozproszenia z podziałem na kategorie (np. typ nadwozia) uwidoczniły zależności między zmiennymi ciągłymi a kategorialnymi. Na przykład, dla zmiennych `horsepower` i `price` zaobserwowano grupowanie się samochodów sportowych w obszarze wysokiej mocy i wysokiej ceny.

Rysunek 27: Wykres rozproszenia mocy silnika i ceny według typu nadwozia
Wykres rozproszenia pokazujący zależność między mocą silnika (hp) a ceną samochodu, z podziałem na typy nadwozia. Widoczne jest grupowanie się poszczególnych kategorii w różnych obszarach wykresu, co wskazuje na zależność ceny i mocy od typu nadwozia.



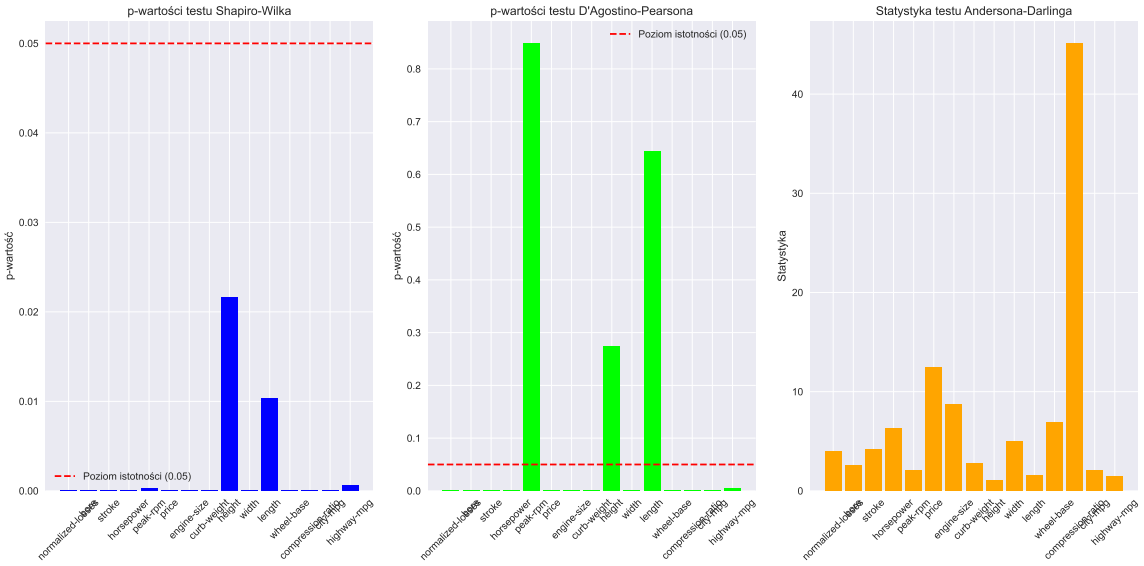
Rysunek 28: Wykresy rozrzutu punktowego według liczby cylindrów
Wykresy rozrzutu punktowego pokazujące rozkład wartości zmiennych numerycznych w zależności od liczby cylindrów silnika. Pozwala to na analizę wpływu liczby cylindrów na inne parametry pojazdu.

5 Testy normalności rozkładów

Aby formalnie ocenić, czy analizowane zmienne podlegają rozkładowi normalnemu, przeprowadzono trzy różne testy statystyczne.

```
shapiro_pvalues = []dagostino_pvalues = []anderson_results = []
for var in continuous_vars : data = df[var].dropna()
Test Shapiro-Wilka ,shapiro_p = stats.shapiro(data)shapiro_pvalues.append(shapiro_p)
Test D'Agostino-Pearsona
    ,dagostino_p = stats.normaltest(data)dagostino_pvalues.append(dagostino_p)
Test Andersona-Darlinga anderson_result = stats.anderson(data, dist =
'norm') anderson_results.append(anderson_result.statistic)
Utworzenie DataFrame z wynikami testw
    normality_tests = pd.DataFrame(    'Variable' : continuous_vars, 'Shapiro_p' :
    shapiro_pvalues, 'DAgostino_p' : dagostino_pvalues, 'Anderson_stat' :
    anderson_results)normality_tests.to_csv('figures/normality_tests.csv', index = False)
Wizualizacja wynikw testw normalnoci fig_norm, axes = plt.subplots(1, 3, figsize = (16, 8))
axes[0].bar(continuous_vars, shapiro_pvalues, color =              '00F')axes[0].axhline(y =
    0.05, color = 'red', linestyle = ' - -', label = 'Poziomistotnoci(0.05)    ')axes[0].set_title('p -
    wartocitestuShapiro - Wilka    ')axes[0].set_ylabel('p -
    warto       ')axes[0].set_xticks(range(len(continuous_vars)))axes[0].set_xticklabels(continuous_vars, rotation =
    45)axes[0].legend()
axes[1].bar(continuous_vars, dagostino_pvalues, color =              '0F0')axes[1].axhline(y =
    0.05, color = 'red', linestyle = ' - -', label = 'Poziomistotnoci(0.05)    ')axes[1].set_title('p -
    wartocitestuDÁgostino - Pearsona    ')axes[1].set_ylabel('p -
    warto       ')axes[1].set_xticks(range(len(continuous_vars)))axes[1].set_xticklabels(continuous_vars, rotation =
    45)axes[1].legend()
axes[2].bar(continuous_vars, anderson_results, color =
            'orange')axes[2].set_title('StatystykatestuAndersona -
    Darlinga')axes[2].set_ylabel('Wartostatystyki       ')axes[2].set_xticks(range(len(continuous_vars)))axes[2].set_xti
    45)
plt.tight_layout()save_fig(fig_norm,          'normality_tests', directory = 'figures')plt.show()
```

Rysunek 29: Kod wykonujący testy normalności: Shapiro-Wilka, D'Agostino-Pearsona i Andersona-Darlinga



Rysunek 30: Wyniki testów normalności dla zmiennych ciągłych

Wykres przedstawia rezultaty trzech testów normalności: Shapiro-Wilka, D'Agostino-Pearsona oraz Andersona-Darlinga. Dla dwóch pierwszych testów pokazano p-wartości (czerwona linia oznacza poziom istotności 0.05), dla trzeciego przedstawiono wartości statystyki testowej (większa wartość oznacza silniejsze odchylenie od rozkładu normalnego).

5.1 Test Shapiro-Wilka

Test ten uważany jest za jeden z najsilniejszych testów normalności, szczególnie dla małych i średnich próbek.

Wyniki testu Shapiro-Wilka wykazały, że dla żadnej z badanych zmiennych p-wartość nie przekroczyła poziomu istotności $= 0.05$, co oznacza, że dla wszystkich zmiennych należy odrzucić hipotezę o normalności rozkładu.

5.2 Test D'Agostino-Pearsona

Ten test łączy informację o skośności i kurtozie w celu oceny normalności.

Wyniki testu D'Agostino-Pearsona wskazały, że tylko dwie zmienne (`horsepower` i `width`) miały p-wartość wyraźnie przekraczającą 0.05, co oznacza brak podstaw do odrzucenia hipotezy o normalności dla tych zmiennych.

5.3 Test Andersona-Darlinga

Test ten kładzie większy nacisk na ogony rozkładu.

Najwyższą statystyką testu Andersona-Darlinga charakteryzowała się zmienna `compression-ratio` (statystyka 45), co wskazuje na znaczne odstępstwo od rozkładu normalnego. Najniższe wartości statystyki zaobserwowano dla zmiennych `height`, `curb-weight` i `engine-size`.

5.4 Wnioski z testów normalności

Na podstawie przeprowadzonych testów można stwierdzić, że:

- Większość zmiennych w zbiorze danych nie pochodzi z rozkładu normalnego.
- Tylko nieliczne zmienne (`horsepower`, `width`) mogą być rozpatrywane jako zbliżone do rozkładu normalnego.
- Przed zastosowaniem metod statystycznych zakładających normalność (np. testy parametryczne) zaleca się przeprowadzenie transformacji danych, np. transformacji logarytmicznej lub standaryzacji.

6 Testy statystyczne dla średniej i wariancji

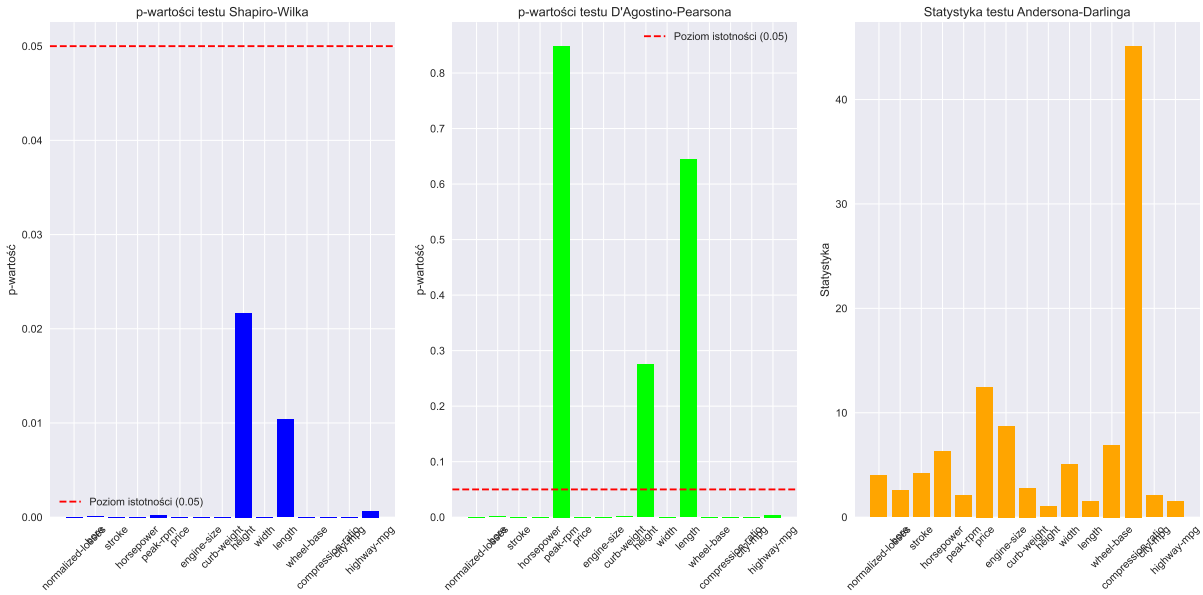
W celu oceny właściwości statystycznych analizowanych zmiennych przeprowadzono kompleksowe testy statystyczne, uwzględniając również normalność rozkładów, która jest kluczowa dla prawidłowej interpretacji wyników.

```
shapiro_pvalues = []dagostino_pvalues = []anderson_results = []
for var in continuous_vars : data = df[var].dropna()
,shapiro_p = stats.shapiro(data)shapiro_pvalues.append(shapiro_p)
,dagostino_p = stats.normaltest(data)dagostino_pvalues.append(dagostino_p)
anderson_result = stats.anderson(data, dist =
'norm') anderson_results.append(anderson_result.statistic)
Create a figure and save a reference to it fig_normality = plt.figure(figsize = (16, 8))
plt.subplot(1, 3, 1) sns.barplot(x=continuous_vars, y = shapiro_pvalues, palette =
    "Blues_d")plt.axhline(y = 0.05, color = 'red', linestyle = '--', label =
    'Poziomistotnoci(0.05)    ')plt.title('p - wartocitestuShapiro - Wilka    ')plt.ylabel('p -
    warto        ')plt.xticks(rotation = 45)plt.legend()
plt.subplot(1, 3, 2) sns.barplot(x=continuous_vars, y = dagostino_pvalues, palette =
    "Greens_d")plt.axhline(y = 0.05, color = 'red', linestyle = '--', label =
    'Poziomistotnoci(0.05)    ')plt.title('p - wartocitestuDÁgostino -
    Pearsona    ')plt.ylabel('p - warto        ')plt.xticks(rotation = 45)plt.legend()
plt.subplot(1, 3, 3) sns.barplot(x=continuous_vars, y = anderson_results, palette =
    "Oranges_d")plt.title('StatystykatestuAndersona -
    Darlinga')plt.ylabel('Statystyka')plt.xticks(rotation = 45)
plt.tight_layout()save_fig(fig_normality, 'normality_tests', directory = 'figures')plt.show()
Test t-Studenta dla redniej t_stats = []p_values_t = []
for var in continuous_vars : data = df[var].dropna()t_stat, p_value =
    stats.ttest_1samp(data, 0)t_stats.append(t_stat)p_values_t.append(p_value)
Create DataFrame with test results stat_tests = pd.DataFrame(    'Variable' :
    continuous_vars, 't_stat' : t_stats, 't_pvalue' : p_values_t)
T-test statistics plot
    fig_tstat = plt.figure(figsize = (12, 6))plt.bar(continuous_vars, t_stats, color =
        'skyblue')plt.axhline(y = 0, color = 'red', linestyle = '--', label = 'Liniaodniesienia(t =
    0)')plt.title('Wynikitestut - Studentadlaredniej    ', fontsize =
    14)plt.ylabel('Statystykat')plt.xticks(rotation =
    45)plt.legend()plt.tight_layout()save_fig(fig_tstat, 't_test_statistics', directory =
    'figures')plt.show()
T-test p-values plot
    fig_tpval = plt.figure(figsize = (12, 6))plt.bar(continuous_vars, p_values_t, color =
        'lightgreen')plt.axhline(y = 0.05, color = 'green', linestyle = '--', label =
    'Poziomistotnoci(0.05)    ')plt.title('p - wartocitestut - Studentadlaredniej        ', fontsize =
    14)plt.ylabel('p - warto        ')plt.xticks(rotation =
    45)plt.yscale('log')Logarytmicznaskaladlalepszejwizualizacjimaychp -
    wartociplt.legend()plt.tight_layout()save_fig(fig_tpval,        't_test_pvalues', directory =
    'figures')plt.show()
Test chi-kwadrat dla wariancji chi2_stats = []p_values_chi2 = []
for var in
    continuous_vars : data = df[var].dropna()n = len(data)dof = n - 1stopnieswobodysample_var =
    np.var(data, ddof = 1)theoretical_var = 1.0zaoonawartoteoretyczna
chi2_stat = dof * sample_var/theoretical_varp_value = 1 - stats.chi2.cdf(chi2_stat, dof)testjednostronny
chi2_stats.append(chi2_stat)p_values_chi2.append(p_value)
Chi-squared test statistics plot
    fig_chi2_stat = plt.figure(figsize = (12, 6))plt.bar(continuous_vars, np.log10(chi2_stats), color =
        'orange')Logarytmicznaskaladlalepszejwizualizacjiplt.title('Statystykatestuchi -
    kwadratdlawariancji(skalalogarytmiczna)', fontsize =
    14)plt.ylabel('log10(Statystykachi - kwadrat)')plt.xticks(rotation =
    45)plt.tight_layout()save_fig(fig_chi2_stat, 'chi2_test_statistics', directory = 'figures')plt.show()
Chi-squared test p-values plot
    fig_chi2_pval = plt.figure(figsize = (12, 6))plt.bar(continuous_vars, p_values_chi2, color =
        'pink')plt.axhline(y = 0.05, color = 'green', linestyle = '--', label =
    'Poziomistotnoci(0.05)    ')plt.title('p - wartocitestuchi - kwadratdlawariancji    ', fontsize =
    14)plt.ylabel('p - warto        ')plt.xticks(rotation =
    45)plt.yscale('log')Logarytmicznaskaladlalepszejwizualizacjimaychp -
    wartociplt.legend()plt.tight_layout()save_fig(fig_chi2_pval,        'chi2_test_pvalues', directory =
    'figures')plt.show()
```

Rysunek 31: Kod wykonujący testy normalności rozkładu oraz testy statystyczne dla średniej i wariancji

6.1 Testy normalności rozkładu

Przed przystąpieniem do testów statystycznych dotyczących średniej i wariancji, istotne jest zbadanie normalności rozkładów analizowanych zmiennych, ponieważ ma to wpływ na wybór odpowiednich metod testowania hipotez.



Rysunek 32: Wyniki testów normalności dla zmiennych ciągłych
Wykres przedstawia wyniki trzech testów normalności: Shapiro-Wilka, D'Agostino-Pearsona oraz Andersona-Darlinga. Czerwona linia przerywana wskazuje poziom istotności 0.05. Wartości p-value powyżej tej linii wskazują na brak podstaw do odrzucenia hipotezy o normalności rozkładu.

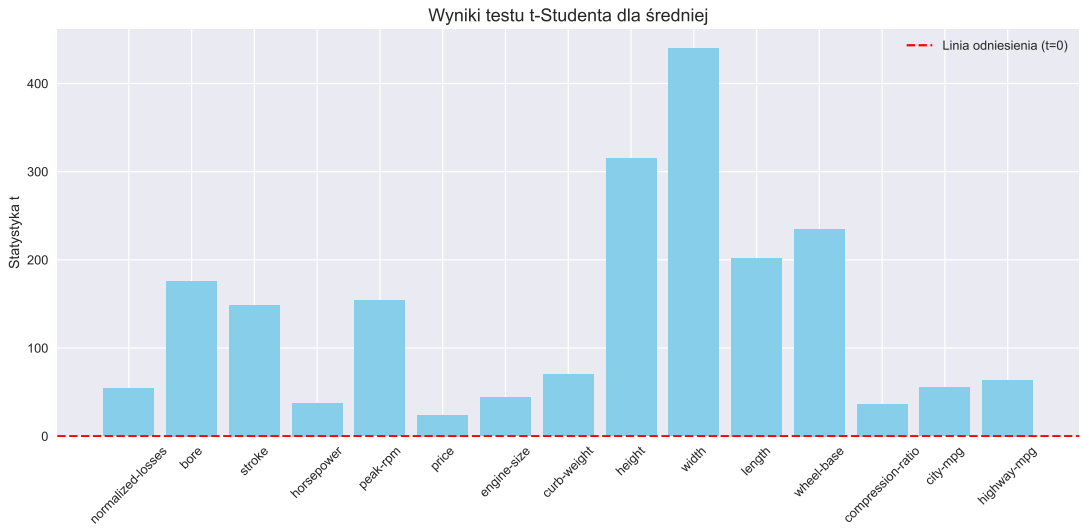
Wyniki testów normalności prowadzą do następujących wniosków:

- Test Shapiro-Wilka wskazuje, że dla żadnej z badanych zmiennych p-wartość nie przekroczyła poziomu istotności 0.05, co oznacza odrzucenie hipotezy o normalności rozkładu dla wszystkich analizowanych zmiennych.
- Test D'Agostino-Pearsona pokazuje, że tylko zmienne `horsepower` i `width` mają p-wartości przekraczające poziom 0.05, co sugeruje, że te dwie zmienne mogą mieć rozkład zbliżony do normalnego.
- Test Andersona-Darlinga, którego wyższa statystyka oznacza większe odstępstwo od normalności, wskazuje na szczególnie silne odchylenia dla zmiennej `compression-ratio` (statystyka ok. 45), a najmniejsze dla zmiennych `height`, `curb-weight` i `engine-size`.

Ze względu na brak normalności rozkładu dla większości zmiennych, wyniki testów parametrycznych przedstawione poniżej należy interpretować z ostrożnością.

6.2 Test t-Studenta dla średniej

Dla każdej zmiennej przeprowadzono jednoetapowy test t-Studenta, weryfikujący hipotezę zerową, że średnia zmiennej wynosi 0.

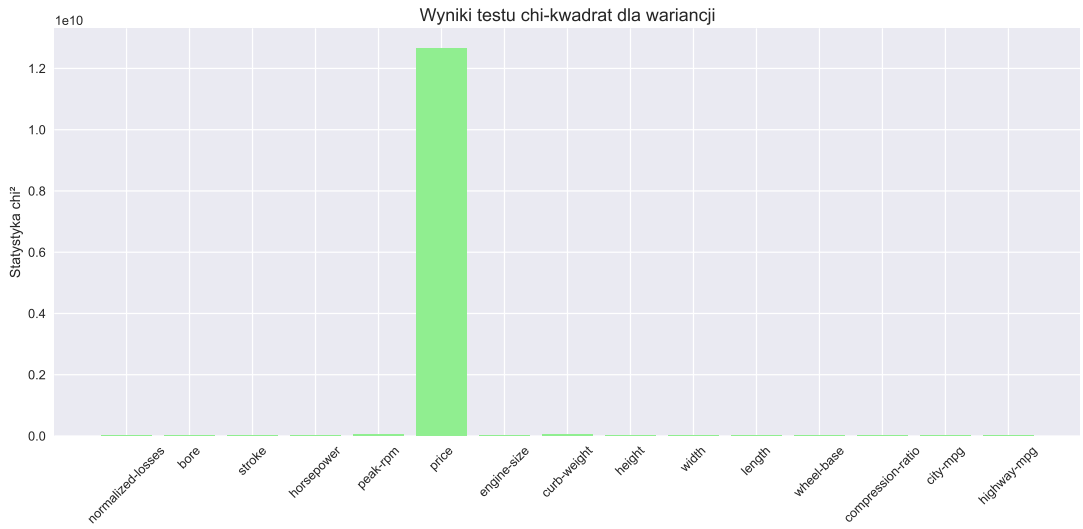


Rysunek 33: Wartości statystyki t-Studenta dla średniej
Wykres przedstawia wartości statystyki t-Studenta dla testu weryfikującego hipotezę zerową, że średnie zmiennych są równe zero. Czerwona linia przerywana oznacza wartość odniesienia (t=0). Wysokie wartości statystyki t dla większości zmiennych wskazują na istotne odchylenia średnich od zera.

Wyniki testu wykazały, że dla wszystkich zmiennych p-wartości były znacznie poniżej poziomu istotności 0.05, co oznacza, że średnie wszystkich zmiennych są istotnie różne od zera. Szczególnie wysokie statystyki t zaobserwowano dla cech takich jak `width`, `height`, `wheel-base` i `length`. Ponieważ jednak większość zmiennych nie ma rozkładu normalnego, wyniki tego testu powinny być traktowane jako przybliżone, zwłaszcza dla zmiennych o znaczącym odchyleniu od normalności.

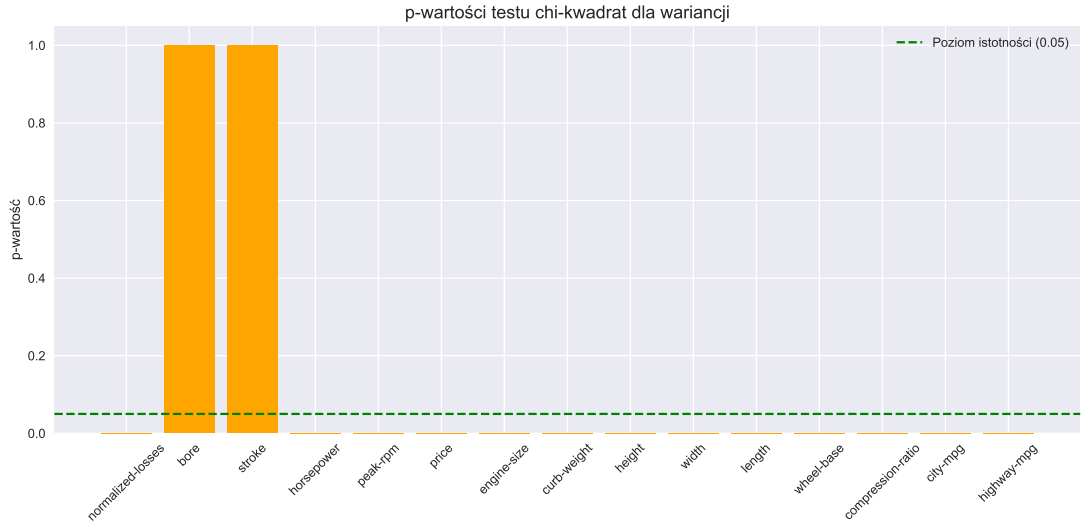
6.3 Test chi-kwadrat dla wariancji

Przeprowadzono również test chi-kwadrat, weryfikujący hipotezę zerową, że wariancja zmiennej jest równa wartości teoretycznej (przyjęto wartość 1 jako odniesienie).



Rysunek 34: Wartości statystyki chi-kwadrat dla wariancji

Wykres przedstawia wartości statystyki chi-kwadrat dla testu wariancji. Szczególnie wyróżnia się zmienna `price` *z bardzo wysoką statystyką, co wskazuje na znaczną zmienność cen samochodów.*



Rysunek 35: p-wartości testu chi-kwadrat dla wariancji

Wykres przedstawia p-wartości testu chi-kwadrat dla wariancji. Zielona linia przerywana oznacza poziom istotności 0.05. Większość zmiennych ma p-wartości poniżej tego poziomu, co oznacza istotną różnicę wariancji od wartości teoretycznej.

Większość zmiennych wykazała p-wartości poniżej poziomu istotności 0.05, co oznacza, że ich wariancje różnią się istotnie od wartości teoretycznej. Wyjątkiem były zmienne `bore` i `stroke`, których p-wartości były bliskie 1, co wskazuje na brak podstaw do odrzucenia hipotezy zerowej. Podobnie jak w przypadku testu t-Studenta, również wyniki testu chi-kwadrat mogą być obarczone błędem ze względu na niespełnienie założenia normalności.

6.4 Interpretacja wyników testów

Na podstawie przeprowadzonych testów możemy sformułować następujące wnioski:

- Większość analizowanych zmiennych nie ma rozkładu normalnego, co potwierdzają wszystkie przeprowadzone testy normalności. Tylko zmienne `horsepower` i `width` mogą być rozpatrywane jako zbliżone do rozkładu normalnego na podstawie testu D'Agostino-Pearsona.

- Mimo niespełnienia założenia normalności, testy parametryczne wykazują, że średnie wszystkich zmiennych są istotnie różne od zera, co potwierdza ich informacyjną wartość w analizach.

- Większość zmiennych charakteryzuje się istotnie różną wariancją od wartości referencyjnej, co może sugerować potrzebę standaryzacji danych przed zastosowaniem niektórych metod analizy.

- Zmienne `bore` i `stroke` wykazują stabilną wariancję, zgodną z wartością teoretyczną.

- Przed zastosowaniem metod statystycznych zakładających normalność (np. testy parametryczne, PCA) zaleca się przeprowadzenie transformacji danych, np. transformacji logarytmicznej lub standaryzacji.

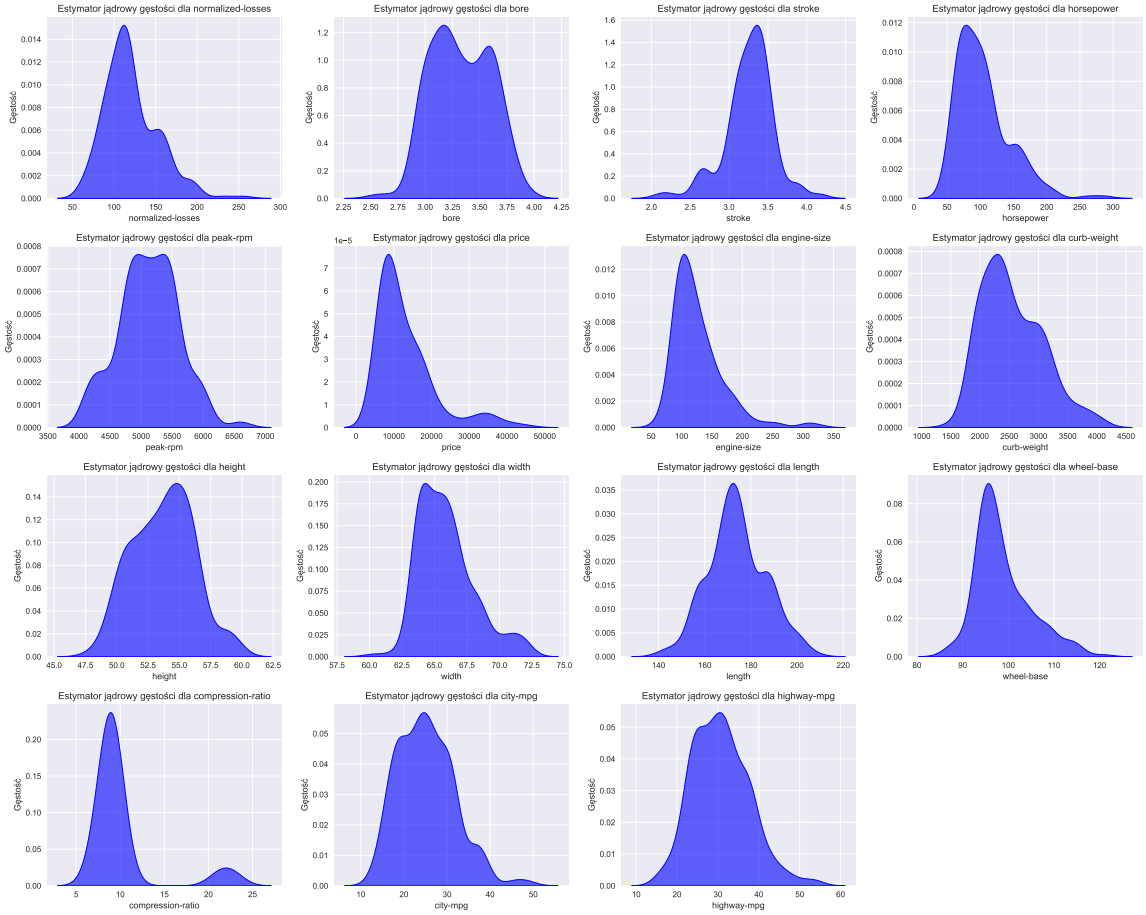
Konieczne jest zatem dalsze postępowanie z ostrożnością przy interpretacji testów parametrycznych i rozważenie zastosowania metod nieparametrycznych jako uzupełnienia analizy dla zmiennych o rozkładach znacząco odbiegających od normalnego.

7 Estymatory jądrowe gęstości

W celu dokładniejszej analizy rozkładów zmiennych zastosowano estymację jądrową gęstości (Kernel Density Estimation, KDE), która umożliwia wygładzenie histogramów i lepsze zrozumienie kształtu rozkładu danych.

```
74 for i, var in
       enumerate(continuous_vars) : if i < len(axes) : sns.kdeplot(df[var], fill = True, color =
75 'purple', alpha=0.7, ax=axes[i])
       axes[i].set_title(f'Rozkad zmiennej var ')axes[i].set_xlabel(var)axes[i].set_ylabel('Gsto ')
76 Ukrycie pustych wykresw for j in range(i+1, len(axes)): axes[j].set_visible(False)
77 plt.tight_layout()save_fig(fig, 'all_kde_plots', directory = 'figures')plt.show()
78 Szczegowy wykres gstoci dla zmiennej compression-ratio (dwumodalny rozkad)
       plt.figure(figsize=(10, 6)) sns.kdeplot(df['compression-ratio'], fill=True,
       color='darkgreen', alpha=0.7) plt.title('Estymator jdrowy gstoci dla zmiennej
       compression-ratio') plt.xlabel('compression-ratio') plt.ylabel('Gsto')
       plt.grid(True, alpha=0.3)
       save_fig(plt.gcf(), 'kde_compression - ratio', directory =
       'figures')plt.show()
79 Wykresy gstoci dla wybranych zmiennych for var in ['price', 'horsepower', 'city-mpg',
       'highway-mpg']: plt.figure(figsize=(10, 6)) sns.kdeplot(df[var], fill=True,
       color='darkblue', alpha=0.7) plt.title(f'Estymator jdrowy gstoci dla zmiennej
       var') plt.xlabel(var) plt.ylabel('Gsto') plt.grid(True, alpha=0.3)
       save_fig(plt.gcf(), f 'kde_var', directory =
       'figures')plt.show()
```

Rysunek 36: Kod tworzący estymatory jądrowe gęstości dla zmiennych ciągłych



Rysunek 37: Estymatory jądrowe gęstości dla wszystkich zmiennych ciągłych
Wizualizacja estymatorów jądrowych gęstości dla wszystkich zmiennych ciągłych. Widoczne są różne kształty rozkładów, od zbliżonych do normalnego, przez silnie asymetryczne, aż po charakterystyczny dwumodalny rozkład zmiennej `compression-ratio`.

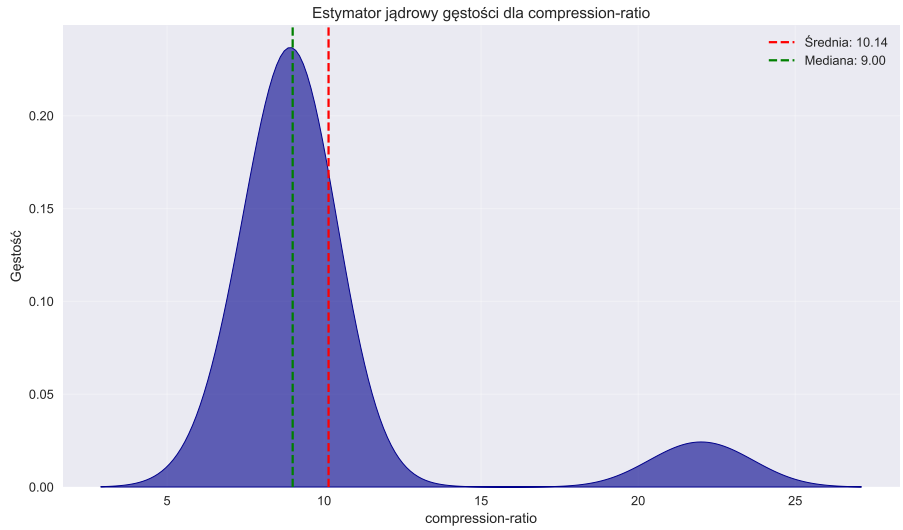
7.1 Analiza rozkładów zmiennych

Estymacja jądrowa gęstości pozwoliła na zaobserwowanie następujących cech rozkładów:

- Wiele zmiennych (`price`, `horsepower`, `engine-size`, `compression-ratio`) charakteryzuje się asymetrycznym rozkładem z długim ogonem po prawej stronie, co sugeruje obecność wartości odstających.
- Niektóre zmienne (`peak-rpm`, `height`, `width`) mają rozkład zbliżony do normalnego, skupiony wokół jednej dominującej wartości.
- Zmienna `compression-ratio` wykazuje rozkład dwumodalny, co może sugerować obecność dwóch różnych klas silników (np. benzynowe vs. wysokoprężne).
- Zmienne `city-mpg` oraz `highway-mpg` mają rozkłady przesunięte ku lewej, co oznacza, że większość samochodów ma umiarkowane zużycie paliwa, a tylko nieliczne cechują się bardzo wysoką efektywnością.

7.2 Znaczenie analizy KDE

Estymatory jądrowe gęstości pozwalają lepiej zrozumieć strukturę danych, wykryć potencjalne problemy oraz dobrać odpowiednie transformacje zmiennych. Analiza ta stanowi istotny etap przygotowawczy przed zastosowaniem algorytmów uczenia maszynowego lub budową modeli statystycznych.



Rysunek 38: Estymator jądrowy gęstości dla zmiennej compression-ratio
Wizualizacja estymacji jądrowej gęstości dla zmiennej compression-ratio, ukazująca charakterystyczny dwumodalny rozkład. Dwa wyraźne szczyty mogą wskazywać na różne kategorie silników.

8 Podsumowanie i wnioski

Przeprowadzona analiza statystyczna zbioru danych Automobile pozwoliła na poznanie kluczowych właściwości opisywanych zmiennych oraz relacji między nimi.

8.1 Główne obserwacje

- Większość zmiennych nie podlega rozkładowi normalnemu, co potwierdziły zarówno wizualizacje, jak i formalne testy statystyczne.
- Zmienne cenowe i techniczne związane z mocą silnika (`price`, `horsepower`, `engine-size`) charakteryzują się silną asymetrią prawostronną, co wskazuje na istnienie niewielkiej liczby drogich, wysokowydajnych pojazdów.
- Zaobserwowano silne korelacje między zmiennymi technicznymi, zgodne z oczekiwaniami inżynieryjnymi (np. dodatnia korelacja między mocą a pojemnością silnika, ujemna korelacja między mocą a zużyciem paliwa).
- Metody bootstrapowe okazały się bardziej odpowiednie do estymacji przedziałowej dla zmiennych o asymetrycznych rozkładach, dostarczając przedziałów ufności lepiej dostosowanych do danych.