Lista 4

Paweł Karwecki, 282249

2025-06-25

Spis treści

1 Zadanie 1

Tak jak na poprzedniej liście, klasyfikacja zostanie przeprowadzona na zbiorze `Vehicle` z pakietu `mlbench`.

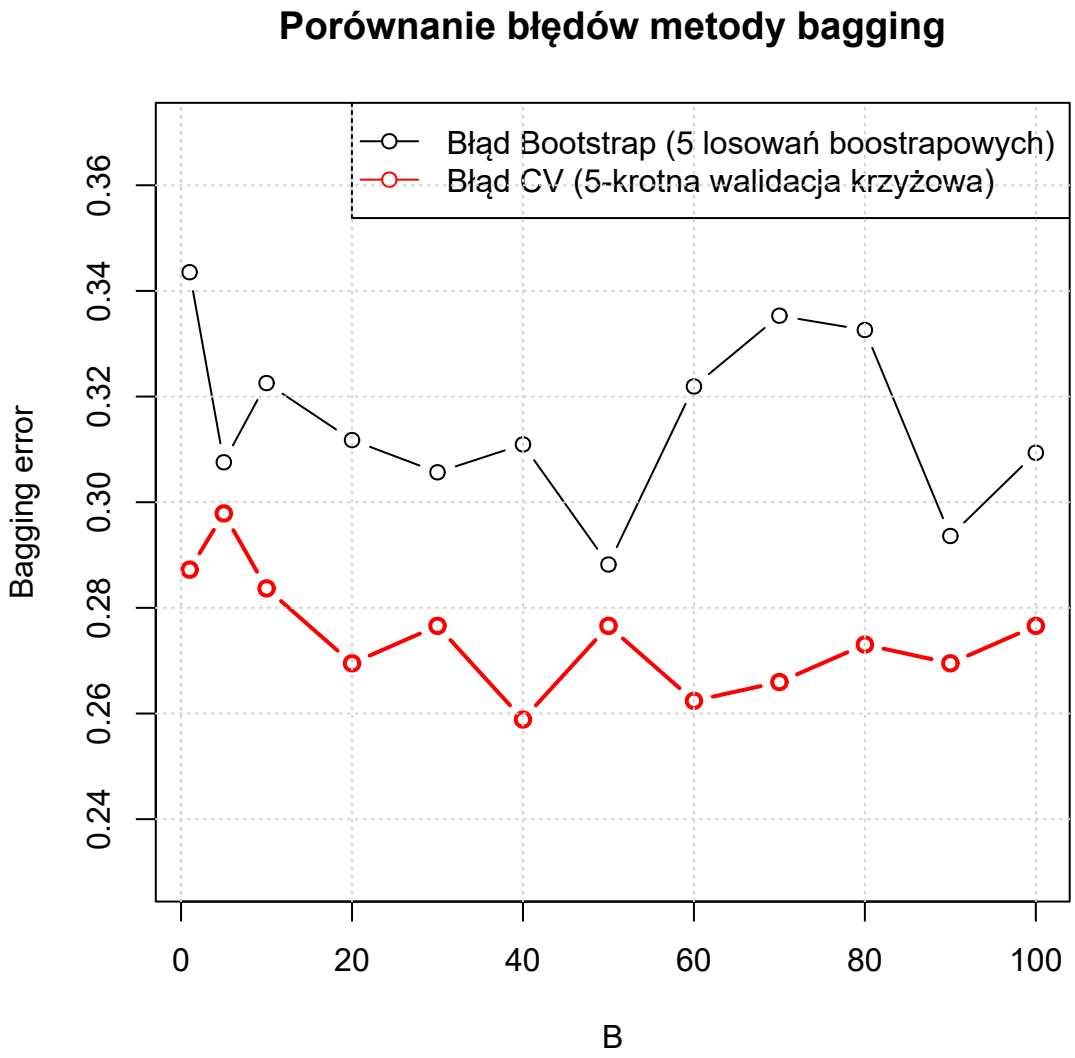
Tabela 1: Rodzaj zmiennych i wytłumaczone nazwy cech

nazwy_zmiennych	typy_zmiennych	opis
Comp	numeric	Zwartość
Circ	numeric	Kolistość
D.Circ	numeric	Kolistość odległościowa
Rad.Ra	numeric	Stosunek promieni
Pr.Axis.Ra	numeric	Stosunek osi głównych
Max.L.Ra	numeric	Maksymalny stosunek długości
Scat.Ra	numeric	Stosunek rozproszenia
Elong	numeric	Wydłużenie
Pr.Axis.Rect	numeric	Prostokątność osi głównych
Max.L.Rect	numeric	Prostokątność maksymalnej długości
Sc.Var.Maxis	numeric	Skalowana wariancja wzdłuż głównej osi
Sc.Var.maxis	numeric	Skalowana wariancja wzdłuż pobocznej osi
Ra.Gyr	numeric	Skalowany promień bezwładności
Skew.Maxis	numeric	Skośność względem głównej osi
Skew.maxis	numeric	Skośność względem pobocznej osi
Kurt.maxis	numeric	Kurtoza względem pobocznej osi
Kurt.Maxis	numeric	Kurtoza względem głównej osi
Holl.Ra	numeric	Stosunek pustek
Class	factor	Typ klasy

1.1 Rodziny klasyfikatorów

W tym podpunkcie wykorzystane zostaną dwie metody rodzin klasyfikatorów: `bagging` i `random forest`. Zbiór uczący i testowy będą takie same, jak na poprzedniej liście.

1.1.1 Metoda bagging



Rysunek 1: Wykres błędów na zbiorze testowym w metodzie bagging w zależności od liczby podzbiorów

Rysunek 1 przedstawia porównanie błędów cross-validation i bootstrap dla metody bagging w zależności od liczby podzbiorów. Najlepsze błędy otrzymujemy dla 20, 40, 60, 70, 90 podzbiorów (biorąc najmniejszy błąd dla każdej metody, w tym przypadku wszystkie błędy są z CV), przyjrzyjmy się bliżej tym błędom.

Tabela 2: Tabela błędów rzeczywistych (na podstawie macierzy pomyłek) na zbiorze testowym wraz z liczbą podzbiorów

Liczba_podzbiorów	Błąd_rzeczywisty
20	0.2659574
40	0.2659574
60	0.2695035
70	0.2517730
90	0.2517730

W tabeli 2 widać, że najlepszy błąd zbioru testowego dla metody bagging otrzymujemy dla podziału zbioru

testowego na 70 lub 90 podzbiorów. Jest to wynik lepszy, niż jakikolwiek inny wynik dla metod klasyfikacji z poprzedniej listy.

1.1.2 Metoda Random Forest

Na początku sprawdźmy najmniejsze średnie błędy (średnia z błędu 5-krotnej walidacji krzyżowej i 5 losowań bootstrapowych) dla Random Forest o różnych wartościach parametrów `mtry` (liczba cech, od 1 do 17) i `ntree` (liczba drzew, od 10 do 100 co 10), biorąc cały zbiór danych `Vehicle`.

Tabela 3: Średnie błędy w zależności od liczby drzew i liczby cech

Liczba_drzew	Liczba_cech	Błąd_cv	Błąd_boot	Średni_Błąd
50	2	0.2364066	0.2388745	0.2376405
90	8	0.2411348	0.2376316	0.2393832
90	2	0.2446809	0.2400794	0.2423801
40	3	0.2411348	0.2445178	0.2428263
70	9	0.2399527	0.2473289	0.2436408

Z tabeli 3 możemy odczytać, że najmniejszy, teoretyczny średni błąd mamy dla 50 drzew i 2 cech. Sprawdźmy teraz błędy na zbiorze testowym, po trenowaniu modelu na zbiorze uczącym.

Tabela 4: Błędy rzeczywiste (na podstawie macierzy pomyłek) na zbiorze testowym w zależności od liczby drzew i liczby cech

Liczba_drzew	Liczba_cech	Błąd_rzeczywisty
20	4	0.2163121
30	4	0.2304965
70	7	0.2304965
20	14	0.2340426
30	14	0.2340426

W tabeli 4 widać, że najmniejszy błąd na zbiorze testowym otrzymujemy dla 20 drzew i 4 cech. Jest to błąd najmniejszy, jaki dotychczas otrzymaliśmy ze wszystkich metod. Warto dodać, że dla tych parametrów, średni błąd teoretyczny (to, co liczyliśmy wcześniej) wynosi 0.249556, a dla 70 drzew i 9 cech otrzymujemy błąd na zbiorze testowym 0.251773.

1.2 Metoda wektorów nośnych SVM

Dla najprostrzego modelu (`kernel = 'linear'`, bez ustalonego kosztu C), stworzonego za pomocą funkcji `svm` z pakietu `e1071` otrzymujemy błąd 0.2056738. Z kolei dla jądra wielomianowego, błąd wynosi 0.2836879, a dla radialnego 0.2553191. Sprawdźmy, czy ustalenie kosztu coś zmieni.

Tabela 5: Błędy metody SVM o różnych jądrach w zależności od poziomu kosztu

Koszt	Błąd_Linear	Błąd_Poly	Błąd_Radial
1	0.2056738	0.2836879	0.2553191
10	0.1950355	0.2340426	0.1879433
50	0.1879433	0.2304965	0.1702128
100	0.1879433	0.2482270	0.1914894
500	0.1950355	0.2234043	0.2092199
1000	0.2021277	0.2340426	0.2163121

Z tabeli 5 widać, że najmniejszy błąd mamy dla kosztu na poziomie 50 i 100 dla jądra liniowego, 500 dla jądra wielomianowego (daje gorszy wynik od liniowego) i 50 dla jądra gaussowskiego (daje najlepszy wynik). Dla większych wartości niż podane, błąd rośnie. Warto przetestować modele o innych stopniach wielomianu w jądrze `polynomial` oraz różnym parametrze gamma w jądrze `radial`. Do wybrania najlepszych parametrów została użyta funkcja `tune()`.

W jądrze wielomianowym najlepszy wynik mamy dla kosztu równego 50 i stopnia 2. Z kolei, dla jądra gaussowskiego najmniejszy błąd mamy dla kosztu na poziomie 1000 i parametru gamma 0.01. Sprawdźmy dokładność tych parametrów na naszych danych:

```
## [1] "Najlepszy błąd w metodzie polynomial: 0.219858156028369"
```

```
## [1] "Najlepszy błąd w metodzie radial: 0.170212765957447"
```

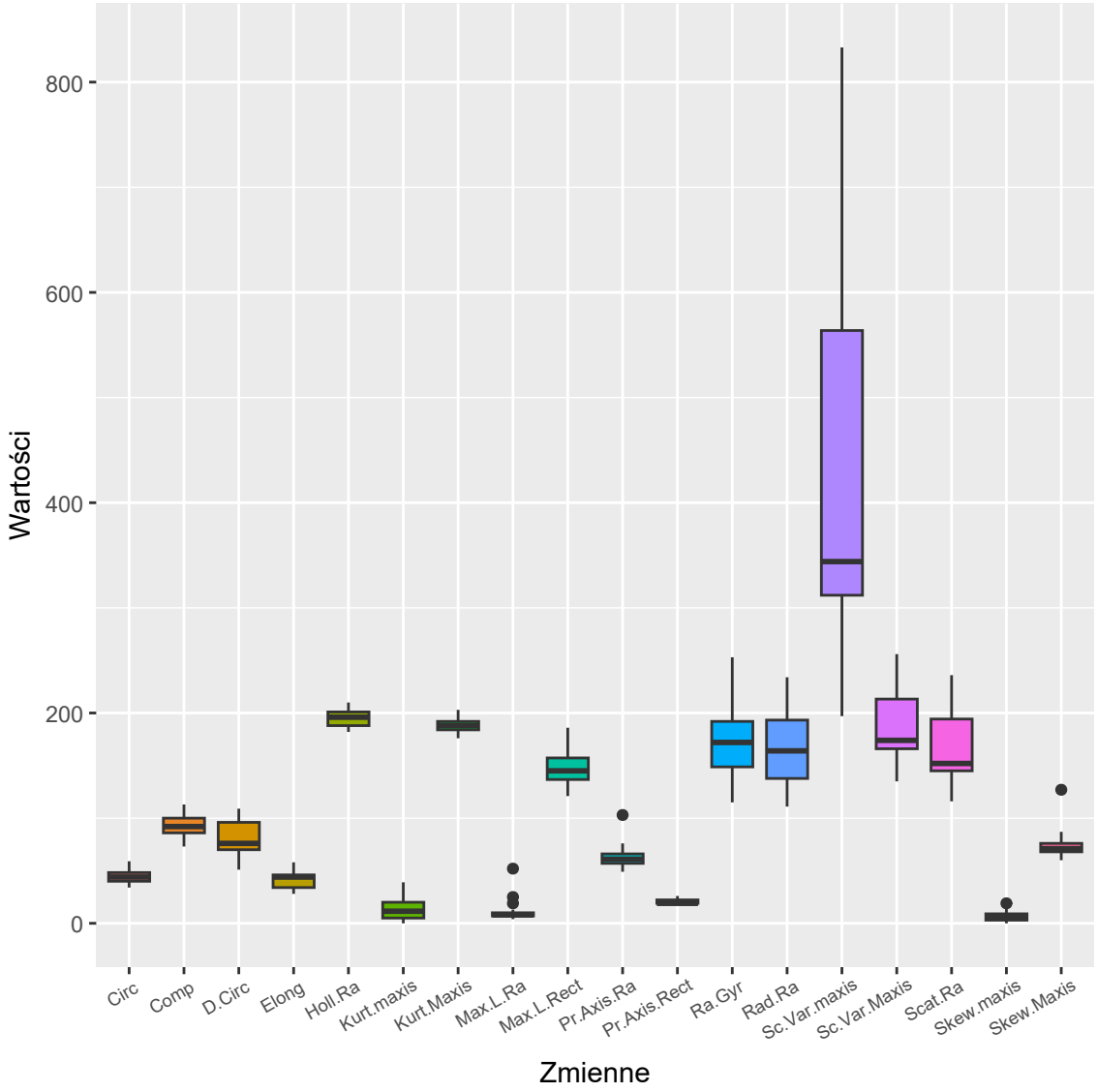
Wyniki pokazują, że jądro wielomianowe, mimo najlepszych parametrów według funkcji `tune()`, ma gorszy błąd niż jądro liniowe. Jądro gaussowskie daje nam najlepszą klasyfikację ze wszystkich przetestowanych metod.

1.3 Porównanie skuteczności metod

Obie metody pozwoliły na zmniejszenie błędu klasyfikacji. Dla metody bagging otrzymaliśmy błąd w okolicy 0.23, dla Random Forest w okolicy 0.21, a dla metody SVM z odpowiednimi parametrami i jądrem gaussowskim udało się otrzymać najlepszy wynik - 0.17. Można zatem stwierdzić, że w przypadku tych danych **SVM ma najlepszą skuteczność**.

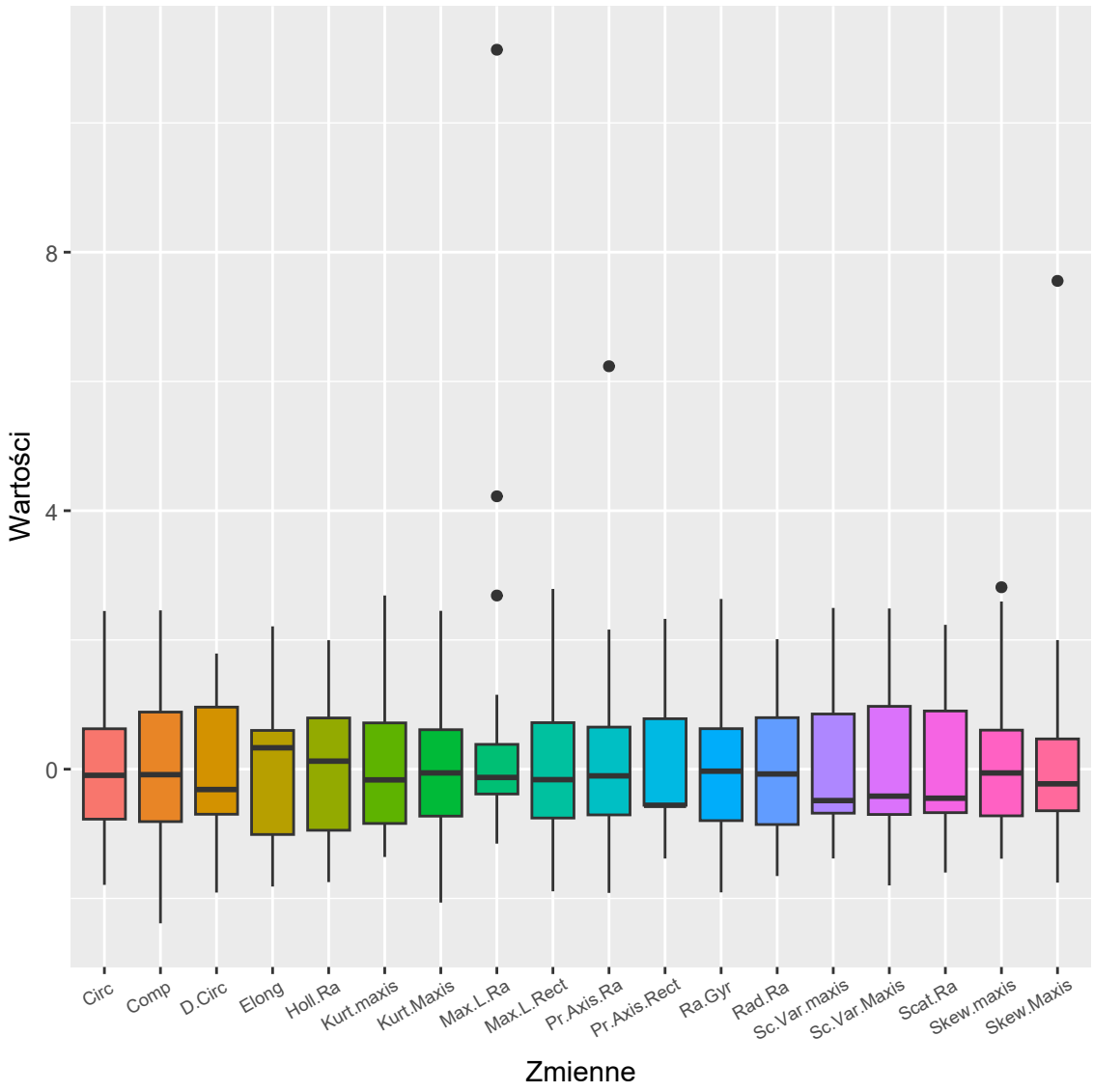
2 Zadanie 2

Do analizy skupień również wykorzystamy dane `Vehicle`.



Rysunek 2: Wykresy pudełkowe poszczególnych zmiennych ze zbioru 200 losowo wybranych obiektów z danych Vehicles

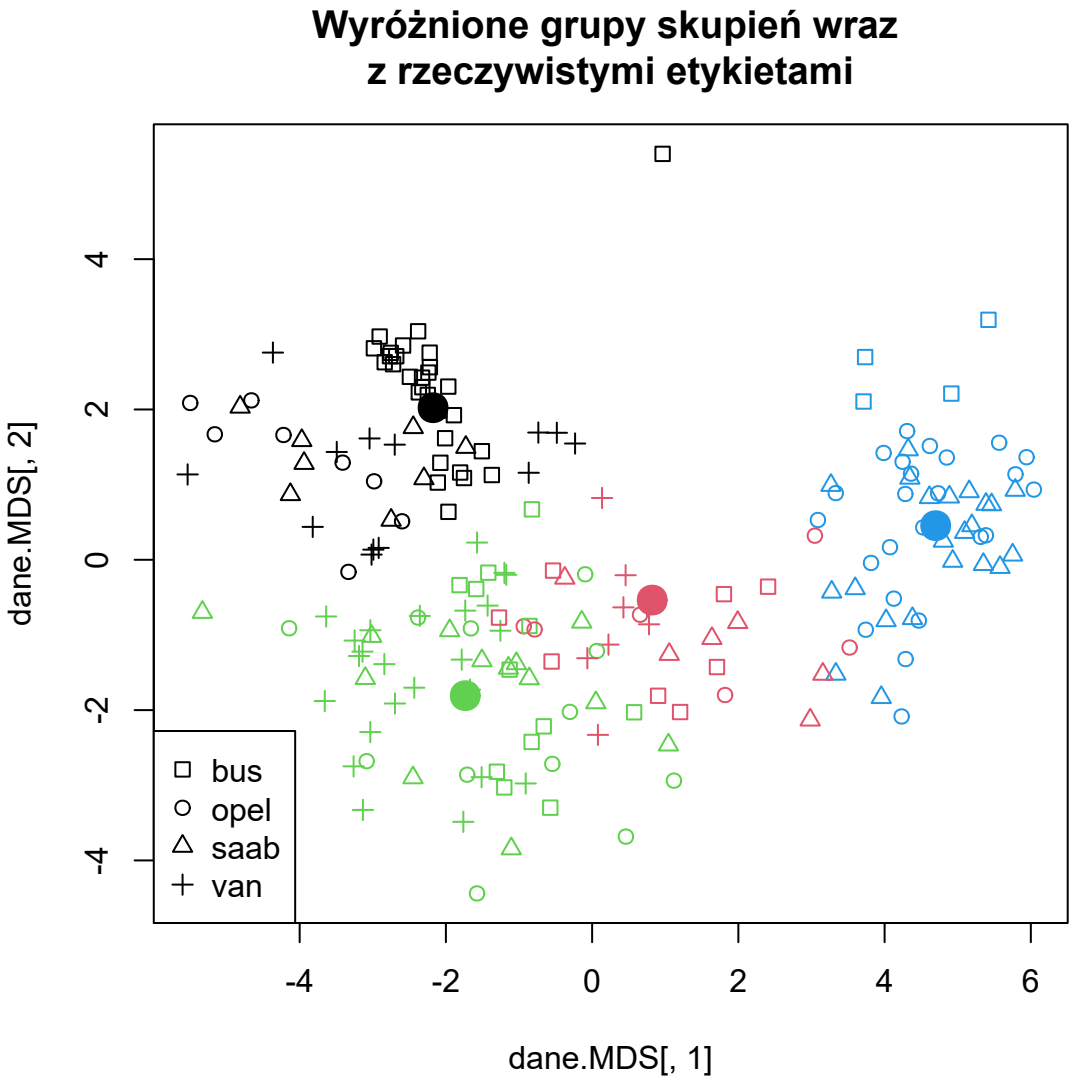
Z rysunku 2 widać, że nasze dane mają bardzo zróżnicowaną wariancję, szczególnie zmienna `Sc.Var.maxis`, więc potrzebna będzie standaryzacja zmiennych.



Rysunek 3: Wykresy pudełkowe poszczególnych zmiennych ze zbioru 200 losowo wybranych obiektów z danych Vehicles po standaryzacji

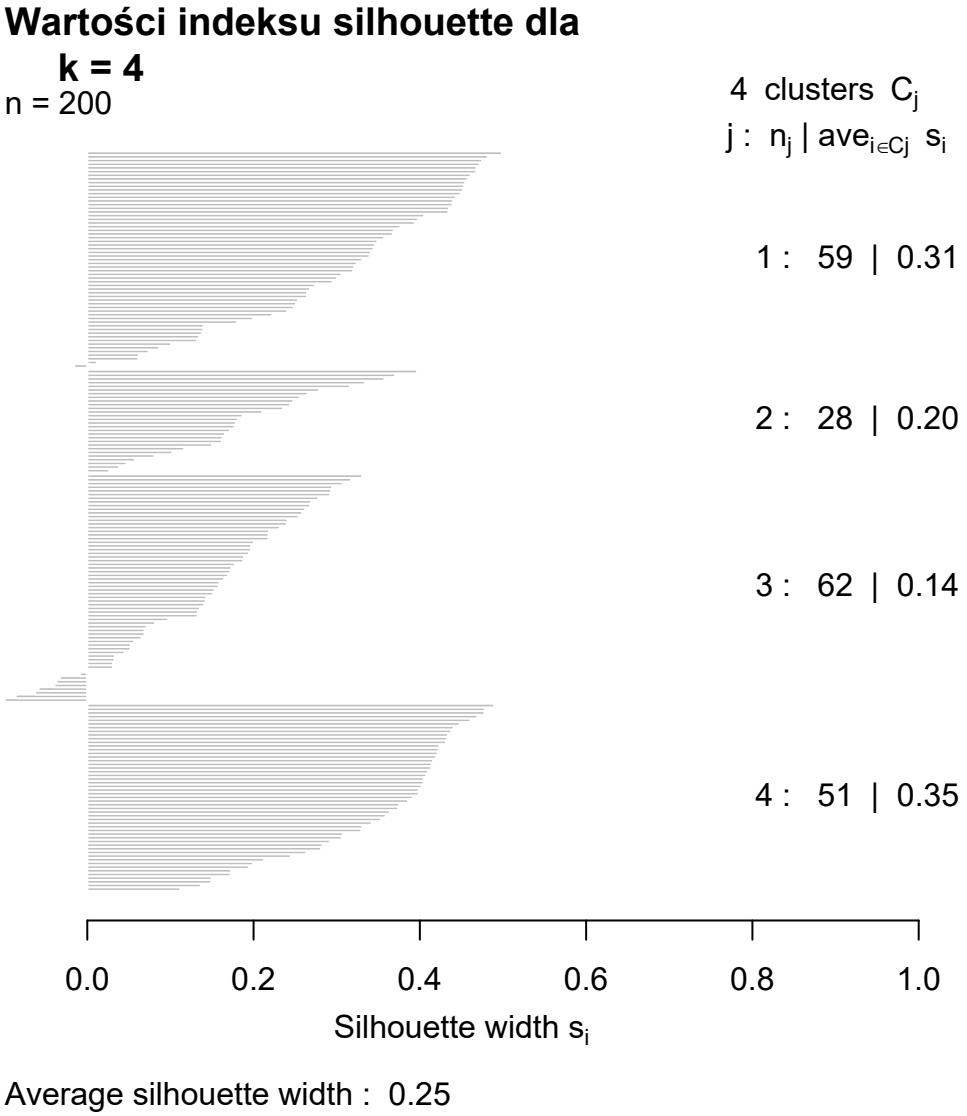
Z rysunku 3 widać, że różnice po standaryzacji już nie są tak wyraźne, jak w poprzednim wypadku.

2.1 Metoda grupująca - algorytm PAM



Rysunek 4: Wizualizacja analizy skupień po zastosowaniu MDS, algorytm PAM

Jak widać na rysunku 4, mamy 4 grupy skupień, wraz z ich medoidami. Zdecydowanie najmniej elementów jest w grupie o czerwonym kolorze. Ogólnie patrząc, dużo obiektów z różnych klas należy do tego samego klastra, np. w zielonej grupie są samochody wszystkich rodzajów. Jedynie w niebieskiej grupie mamy tylko 3 rodzaje pojazdów - `bus`, `opel`, i `saab`. Widać również, że grupa zielona i czerwona na siebie nachodzą. Sprawdźmy, jak prezentują się wartości indeksu silhouette.



Rysunek 5: Wartości indeksu silhouette

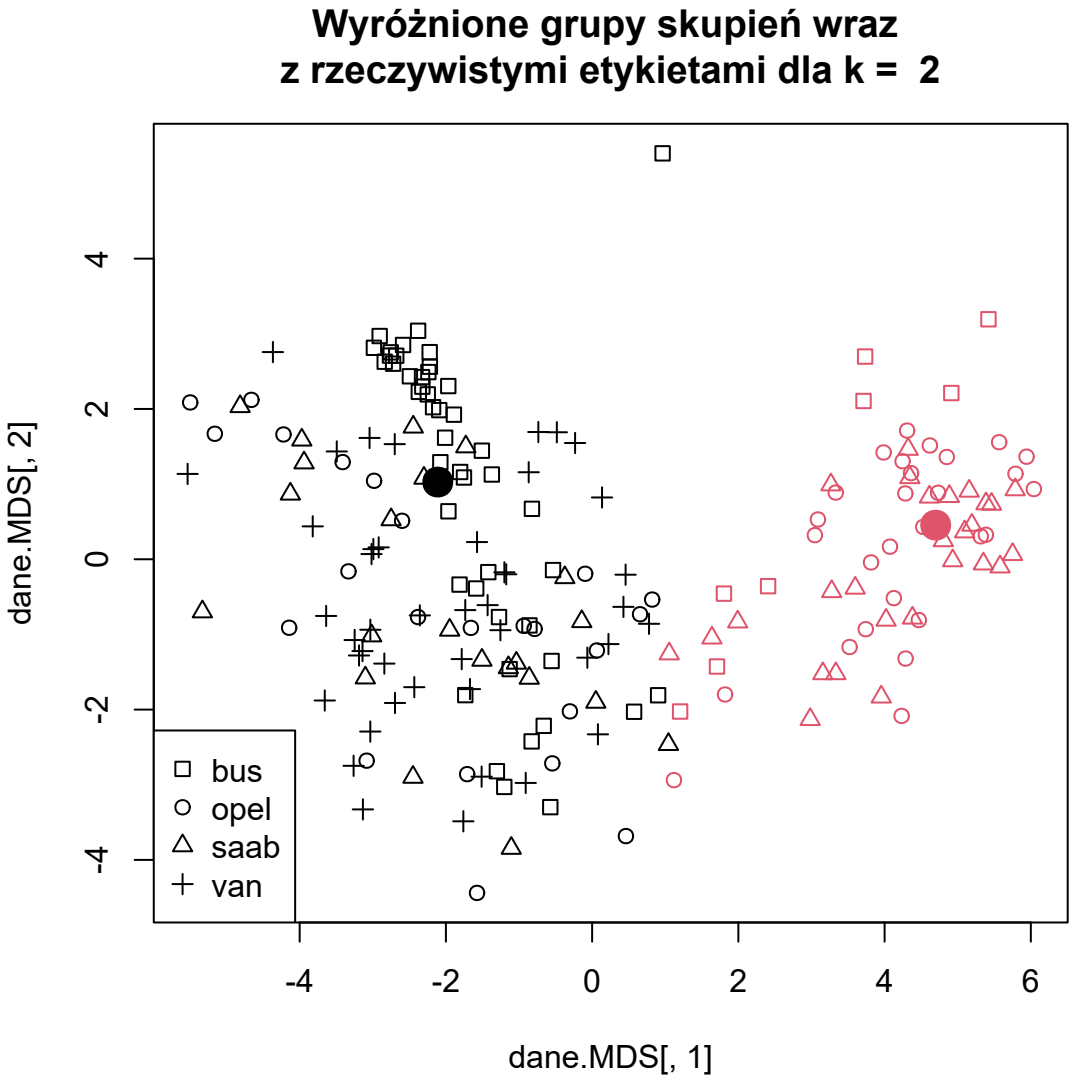
Na rysunku 5 widać, że najlepsza grupa ma średnią indeksu na poziomie 0.35, a ogólna średnia to 0.25. Wynik ten nie jest najlepszy, szczególnie gdy spojrzymy na 3. klasę, w której kilka obiektów ma ujemną wartość indeksu, co tylko podkreśla problem poprawnego przyporządkowania do dobrego skupienia. Przyjrzyjmy się, co zwraca miara zgodności partycji (funkcja `matchClasses`)

```
## Direct agreement: 1 of 4 pairs
## Iterations for permutation matching: 6
## Cases in matched pairs: 42.5 %
```

```
##      1      2      3      4
##  "bus" "saab"  "van" "opel"
```

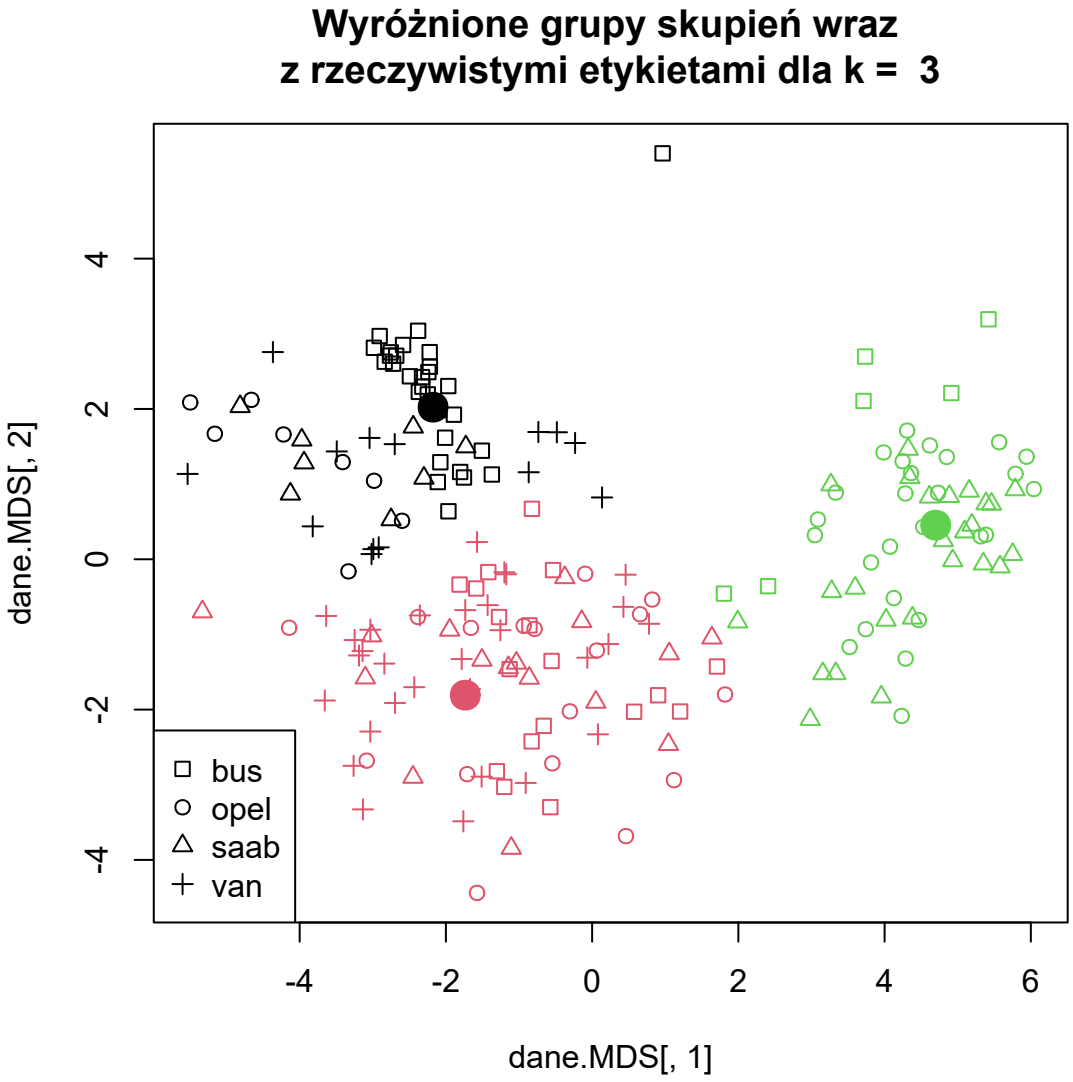
Jak widać, udało się poprawnie przyporządkować jedynie 42.5% przypadków do poszczególnych klas. Warto podkreślić, że jest to wynik przy ustawieniu argumentu `method` na `exact`, czyli wymuszone zostało użycie wszystkich czterech klas.

Przyjrzyjmy się teraz innej liczbie skupień (K=2 i K=3).



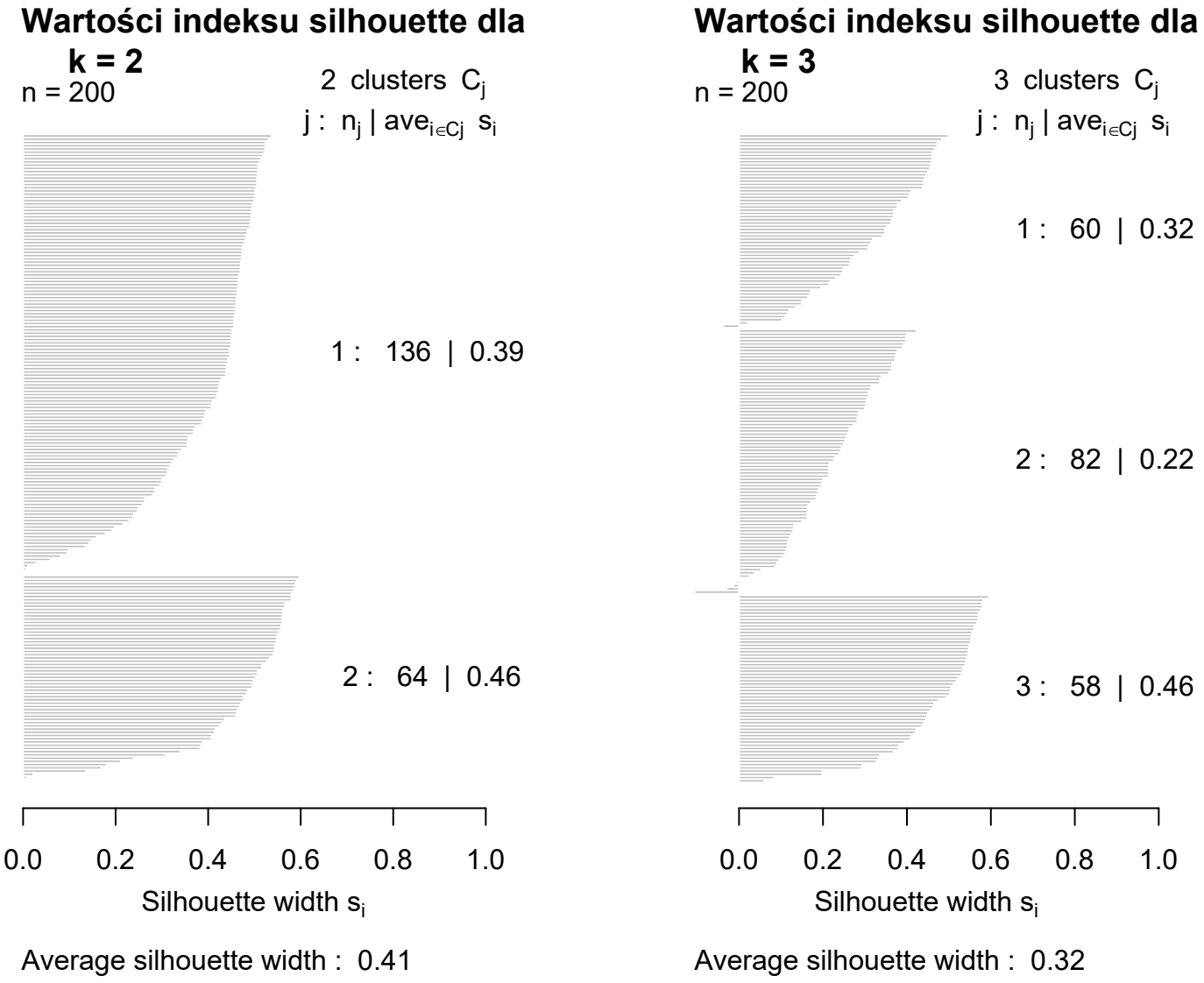
Rysunek 6: Wizualizacja analizy skupień po zastosowaniu MDS przy liczbie klastrów k = 2

Jak widać na rysunku 6, duża część obiektów została przypisana do klastra o kolorze czarnym. Warto też zauważyć, że wszystkie obiekty z klasy `van` są w czarnym klastrze.



Rysunek 7: Wizualizacja analizy skupień po zastosowaniu MDS przy liczbie klastrów k = 3

Z kolei na rysunku 7 liczność obiektów w danej klasie wydaje się być podobna. Samochody klasy `van` występują jedynie w czerwonym i czarnym klastrze. Zauważalna jest również mała ilość pojazdów typu `bus` w zielonym klastrze.



Rysunek 8: Wartości indeksu silhouette przy liczbie klastrów k = 2 i k = 3

Wykresy z rysunku 8 pokazują, że zmniejszenie liczby klastrów przynosi lepsze wyniki. Wszystkie obiekty mają dodatnie wartości indeksu, a średnia dla k = 2 jest 1.64 razy większa, niż w przypadku k = 4.

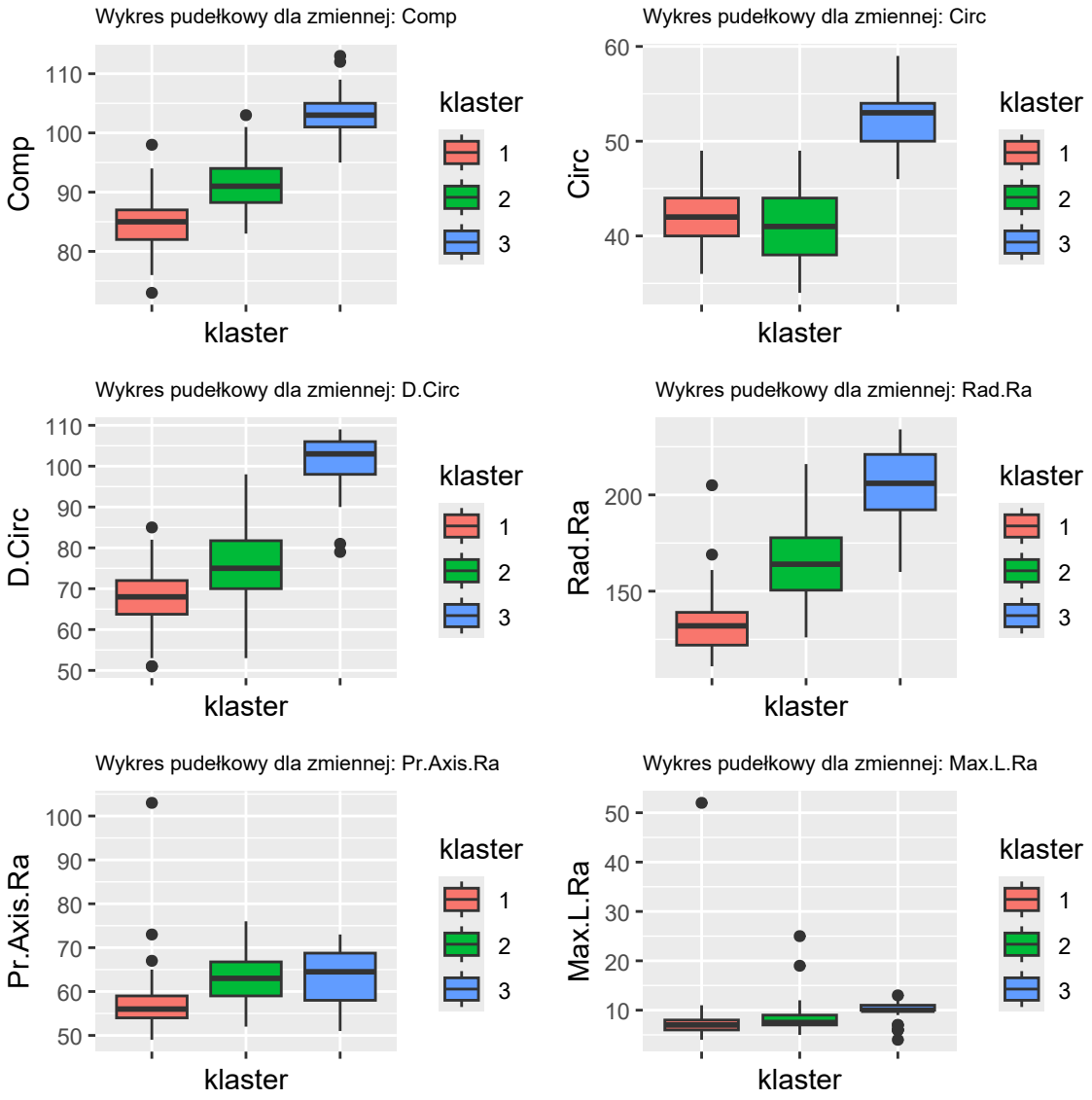
```
## Cases in matched pairs: 38 %

##      1      2
##  "bus" "opel"

## Cases in matched pairs: 43.5 %

##      1      2      3
##  "bus"  "van" "opel"
```

Miara zgodności partycji zwraca lepszy wynik dla k = 3. Mimo lepszego indeksu silhouette dla k = 2, skuteczność grupowania jest gorsza i wynosi jedynie 38%.

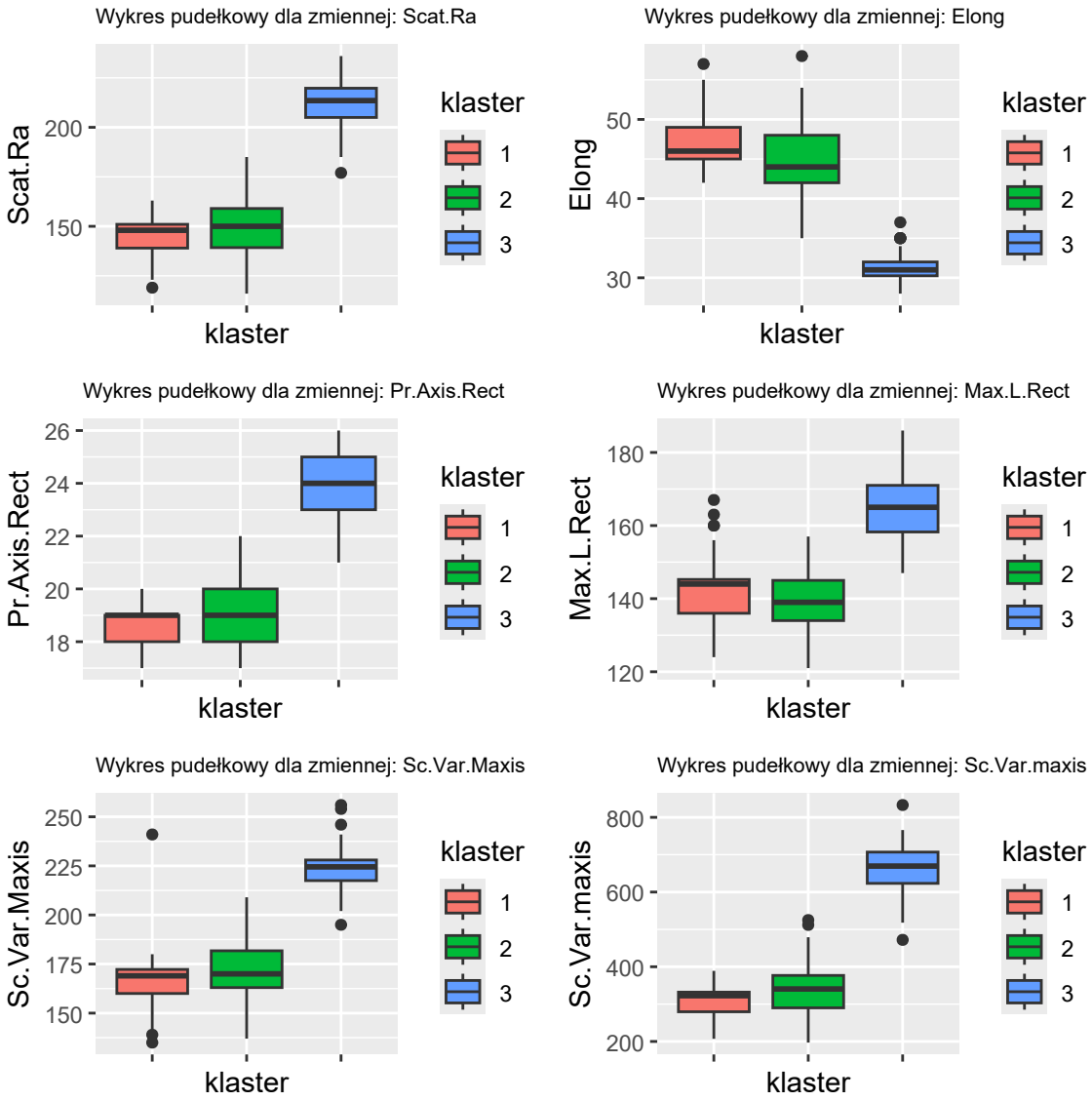
Podsumowując, dla k = 3 mamy większy indeks silhouette i lepszą skuteczność grupowania niż dla k = 4, więc w algorytmie PAM to tę **liczbę klastrów można uznać za najlepszą**.

Tabela 6: Średnie wartości cech obiektów z danych klastrów, algorytm PAM

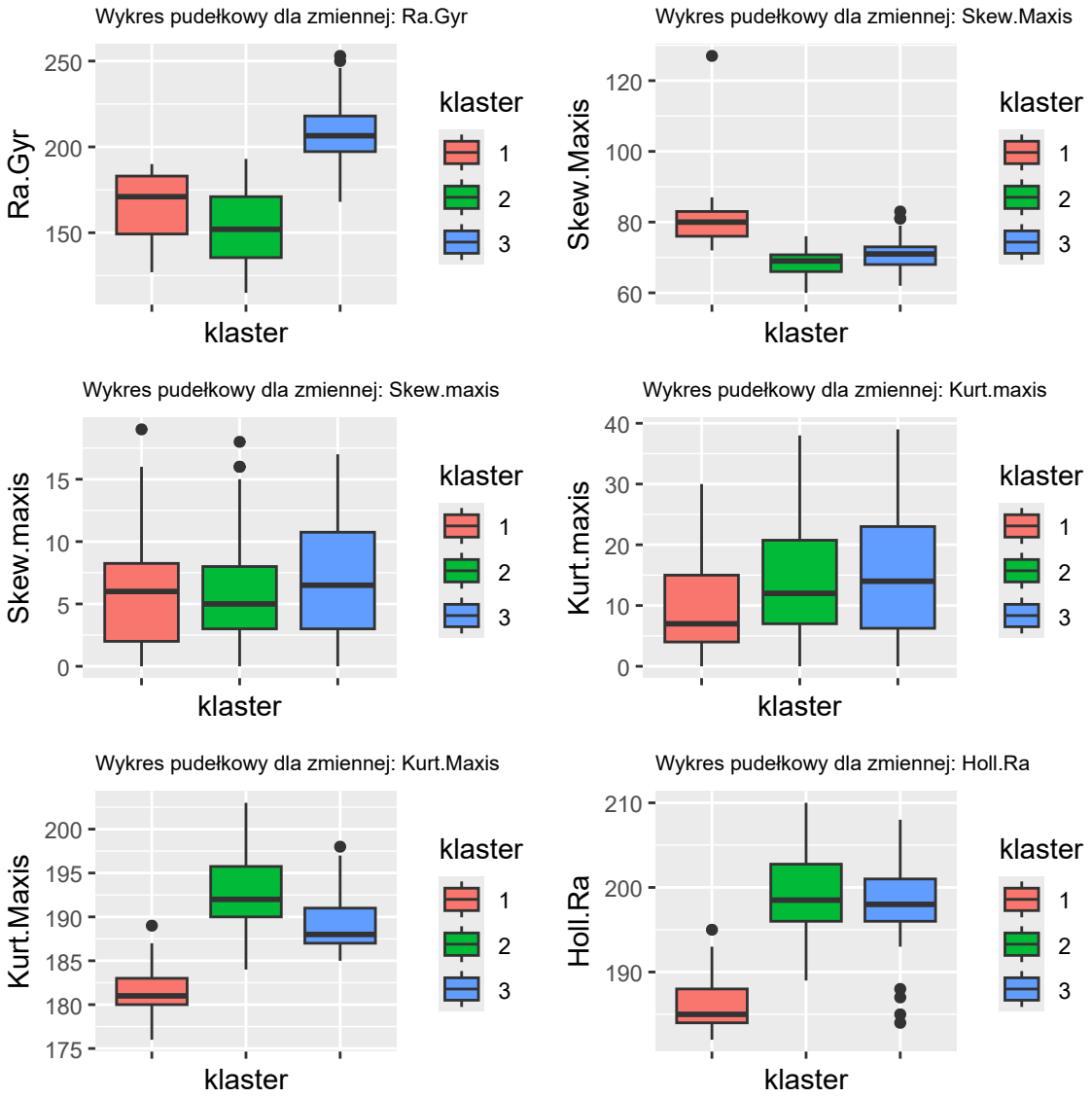
	1	2	3
Comp	85.016667	91.085366	102.931034
Circ	42.066667	41.024390	52.137931
D.Circ	67.550000	76.182927	101.500000
Rad.Ra	132.100000	164.841463	204.396552
Pr.Axis.Ra	57.733333	63.268293	63.534483
Max.L.Ra	7.716667	8.000000	10.017241
Scat.Ra	145.050000	150.158537	210.275862
Elong	46.933333	44.573171	31.637931
Pr.Axis.Rect	18.750000	19.085366	23.810345
Max.L.Rect	141.533333	139.317073	164.379310
Sc.Var.Maxis	166.533333	172.975610	223.827586
Sc.Var.maxis	308.900000	341.597561	658.413793
Ra.Gyr	165.300000	153.524390	208.293103
Skew.Maxis	80.333333	68.243902	70.862069
Skew.maxis	6.100000	5.853658	7.017241
Kurt.maxis	9.533333	13.939024	15.586207
Kurt.Maxis	181.466667	192.841463	189.103448
Holl.Ra	186.016667	199.609756	198.017241



Rysunek 9: Wykresy pudełkowe zmiennych z podziałem na klastry, algorytm PAM



Rysunek 10: Wykresy pudełkowe zmiennych z podziałem na klastry, algorytm PAM



Rysunek 11: Wykresy pudełkowe zmiennych z podziałem na klastry, algorytm PAM

Z tabeli 6 i rysunków 9-11 widać, że cechy `Comp`, `D.Circ`, `Rad.Ra`, `Sc.Var.maxis` oraz `Ra.Gyr` najlepiej pozwalają na rożnienie klastra. Klaster pierwszy wyróżnia się wysoką wartością `Skew.Maxis` oraz niską wartością w `Pr.Axis.Ra` oraz ostatnich trzech cechach. Dla pozostałych atrybutów, wartością wyróżnia się klaster trzeci. Poza wspomnianymi na początku pięcioma cechami, wartości w drugim klastrze często są zbliżone do wartości pierwszej lub trzeciej grupy.

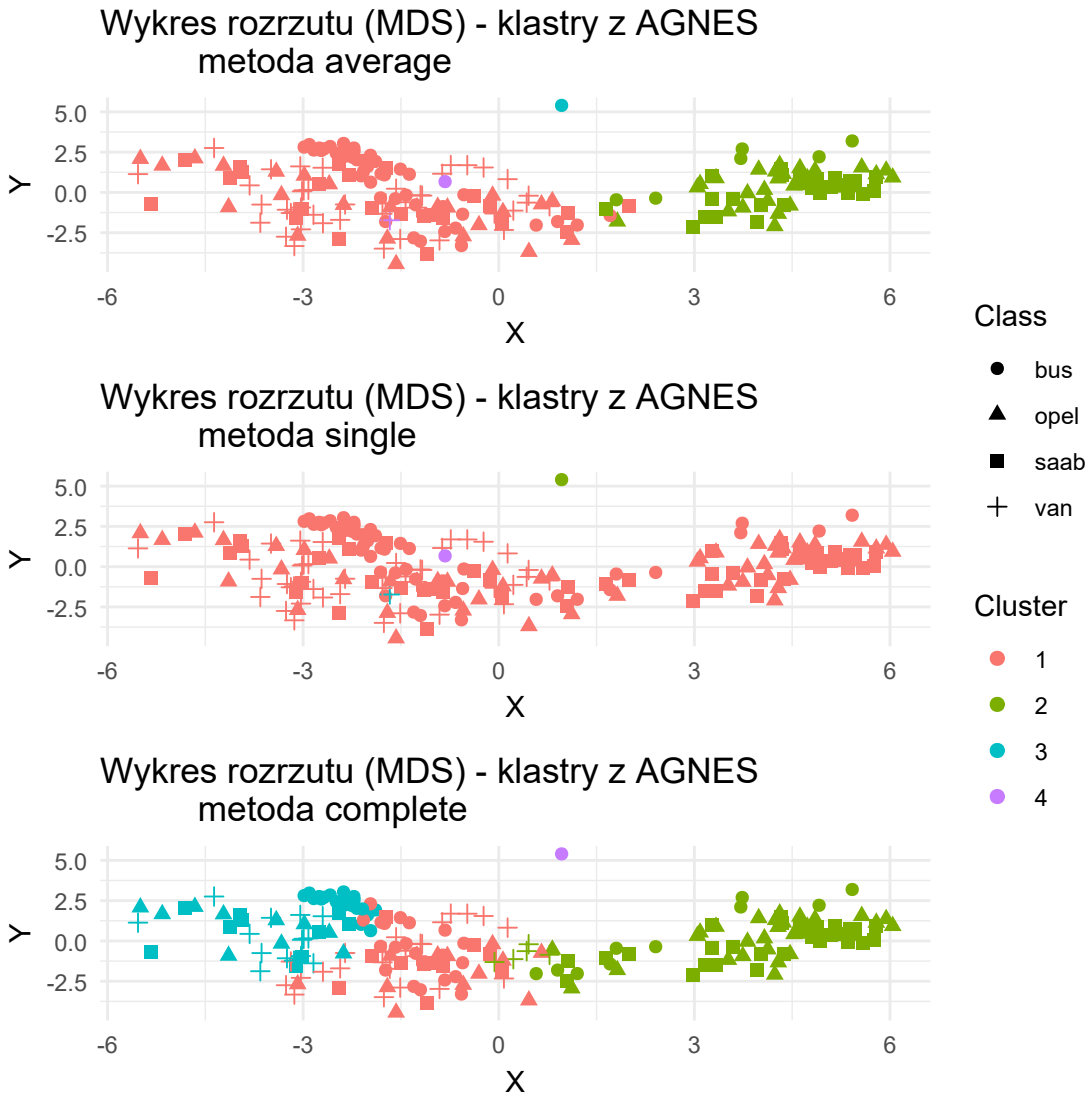
Tabela 7: Wartości cech medoidów z danych klastrów

	1	2	3
Comp	86	88	104
Circ	43	39	55
D.Circ	66	70	105
Rad.Ra	130	166	216
Pr.Axis.Ra	56	66	68
Max.L.Ra	7	7	11
Scat.Ra	152	148	205
Elong	44	44	32
Pr.Axis.Rect	19	19	23
Max.L.Rect	142	134	169
Sc.Var.Maxis	177	167	221
Sc.Var.maxis	340	332	623
Ra.Gyr	173	143	216
Skew.Maxis	81	69	71
Skew.maxis	6	5	9
Kurt.maxis	14	13	18
Kurt.Maxis	181	193	189
Holl.Ra	185	201	196

Tabela 7 przedstawia wartości medoidów danych klastrów. Wyróżnia się tutaj na pewno wysoka chociażby wartość `Sc.Var.maxis`, `Rad.Ra` i `Scat.Ra` w 3. klastrze czy też niskie wartości pierwszego klastra w `Rad.Ra`, `Holl.Ra` i `Pr.Axis.Ra`.

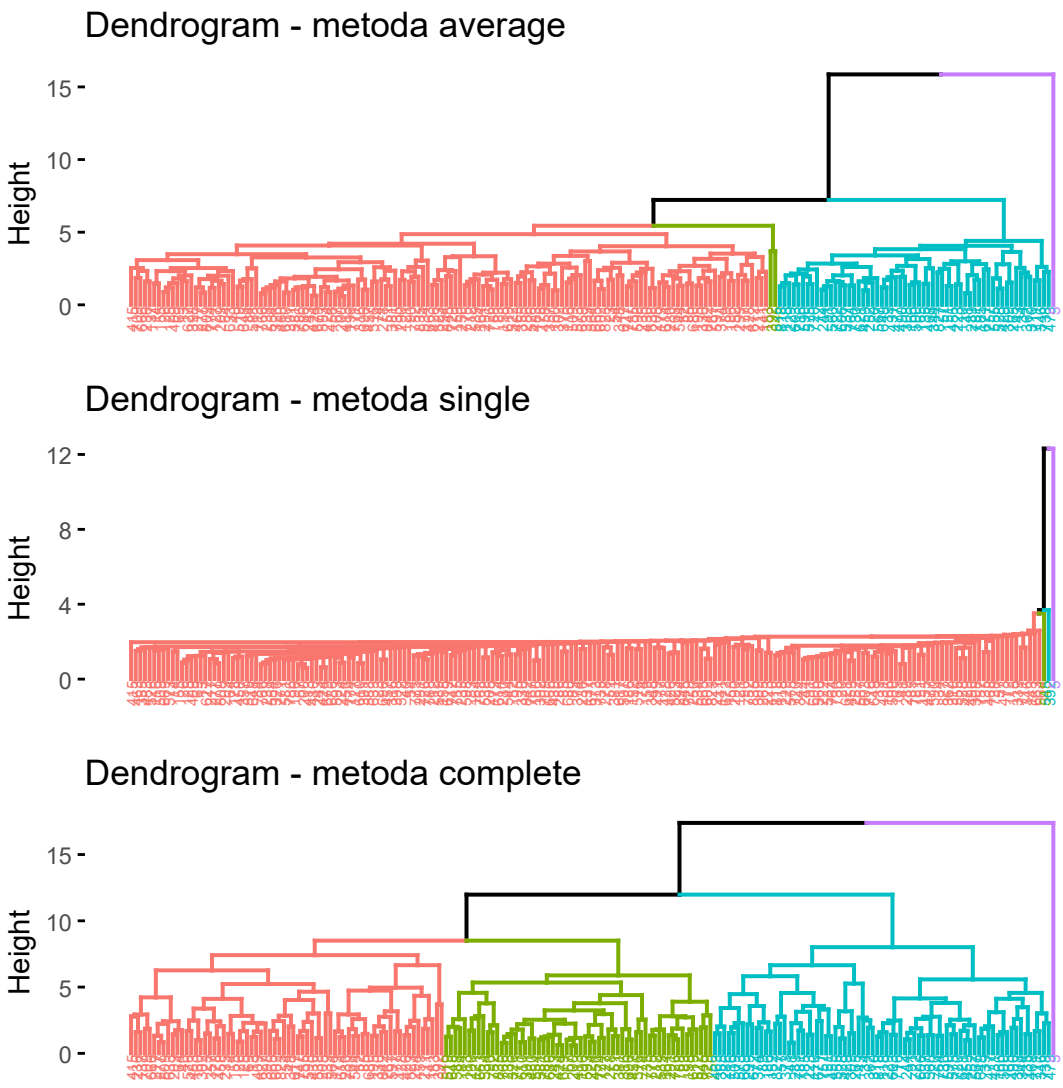
2.2 Metoda hierarchiczna - algorytm AGNES

W algorytmie AGNES skorzystamy z trzech metod łączenia klastrów: `average` (średnia odległość), `complete` (najdalszy sąsiad), `single` (najbliższy sąsiad).



Rysunek 12: Wizualizacja analizy skupień po zastoswaniu MDS, algorytm AGNES, k = 4

Na rysunku 12 widoczne są problemy w łączeniu klastrów w metodzie `average` i `single`. Duża część obiektów przypisywana jest do czerwonego skupienia, a skupienia o kolorze fioletowym i niebieskim mają jedynie 1 albo 2 elementy. Najbardziej różnorodne skupienia są w przypadku metody `complete`, aczkolwiek tutaj również w skupieniu fioletowym jest tylko jeden rekord. Spójrzmy na dendrogramy tych metod.



Rysunek 13: Dendrogramy metody AGNES dla k = 4

Rysunek 13 tylko potwierdza problem w podziale grupy na klastry, szczególnie w przypadku metody `single`. Warto dodać, że kolory z tego rysunku nie pokrywają się z kolorami z rysunku 12 (zielony kolor jest zamieniony z niebieskim), ale struktura podziału jest zachowana. Przejdźmy do wartości wskaźników wewnętrznych i zewnętrznych.

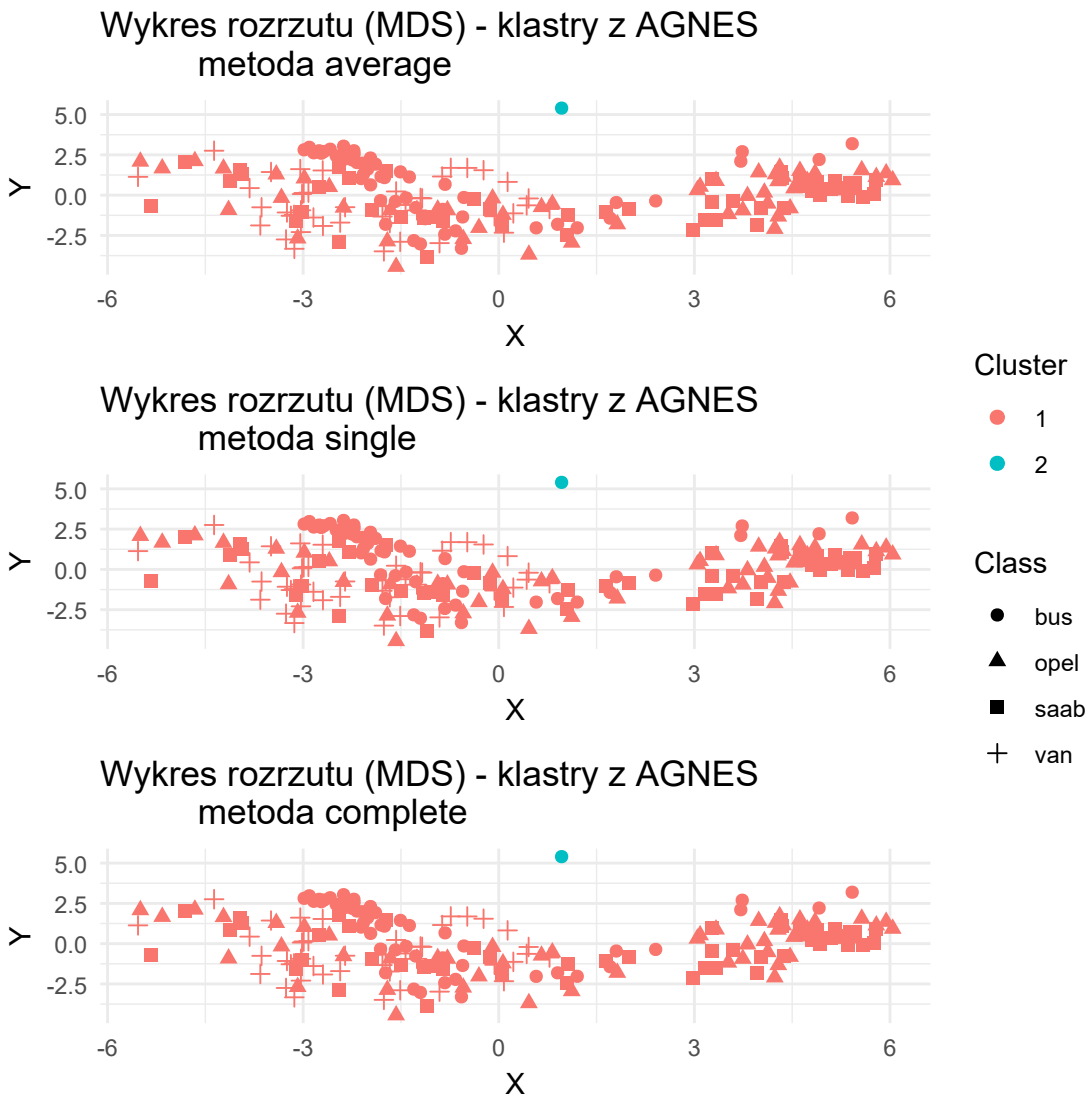
Tabela 8: Porównanie średnich indeksów silhouette w klastrach dla różnych metod łączenia klastrów (k = 4)

klastr	average	single	complete
1	0.2198659	0.0113611	0.1647034
2	0.4892926	0.0000000	0.3074589
3	0.0000000	0.0000000	0.2821031
4	0.3181458	0.0000000	0.0000000

Z tabeli 8 możemy odczytać, że najlepszą wartość indeksu w klastrach 1,2,4 mamy dla metody `average`, a w klastrze 3 najlepiej prezentuje się `complete`. Warto również zauważyć, że średnia wartość indeksu silhouette

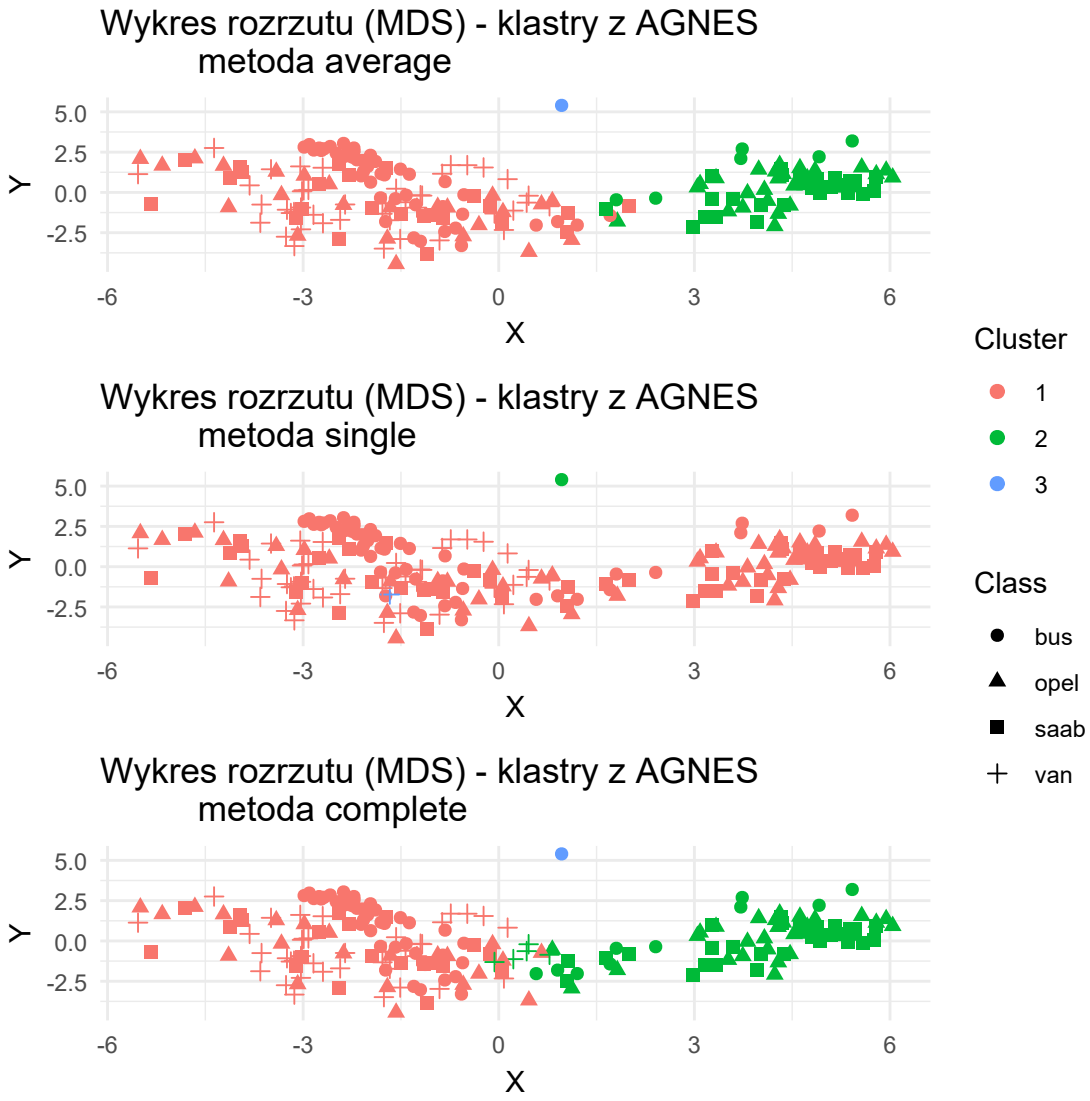
dla `average` to 0.2568261, czyli więcej niż 0.1885663 w metodzie `complete`. Jest to o tyle ciekawe, że w metodzie średniej odległości ponad połowa rekordów została przypisana do jednego klastra.

Za pomocą funkcji `matchClasses()` wiemy, że dla najlepsze dopasowanie mamy dla `complete` i wynosi ono 38.5%, trochę gorzej wypada `average` z wynikiem 38%, a najgorzej (ponownie) wypada `single` z wynikiem 27%. Sprawdźmy teraz, co się stanie, gdy zmniejszymy liczbę sąsiadów.



Rysunek 14: Wizualizacja analizy skupień po zastoswaniu MDS, algorytm AGNES, k = 2

Na rysunku 14 wszystkie metody zwróciły ten sam wynik, który oczywiście nie oddaje dobrze faktycznego stanu klas w naszych danych. Wychodzi na to, że spośród 200 samochodów z 4 klas, jedynie jeden nie pasuje do reszty. Co prawda średni wskaźnik indeksu silhouette wynosi 0.3289616, ale zgodność klas to jedynie 27.5%. Zdecydowanie nie jest to najlepsza liczba klastrów.



Rysunek 15: Wizualizacja analizy skupień po zastoswaniu MDS, algorytm AGNES, k = 3

Dla k = 3 wyniki już prezentują się ciekawiej, co widać na rysunku 15. W metodzie `average` i `complete` mamy ewidentny podział na 2 grupy, z jednym, punktowym wyjątkiem (jest to ten sam wyjątek, co w przypadku k = 2). Metoda `single` zwraca bardzo podobny wynik jak dla k = 2, ale mamy jeden obiekt z klasy `van` w innym skupieniu.

Tabela 9: Porównanie średnich indeksów silhouette w klastrach dla różnych metod łączenia klastrów (k = 4)

klastr	average	single	complete
1	0.3984276	0.162343	0.4020151
2	0.4900863	0.000000	0.3803370
3	0.0000000	0.000000	0.0000000

Z tabeli 9 widać, że w przypadku `average` i `complete` mamy najlepsze wartości indeksów. Ich średnie są na poziomie 0.2961713 i 0.260784, ale wynik ten jest oczywiście zaniżony przez zera w ostatnim wierszu

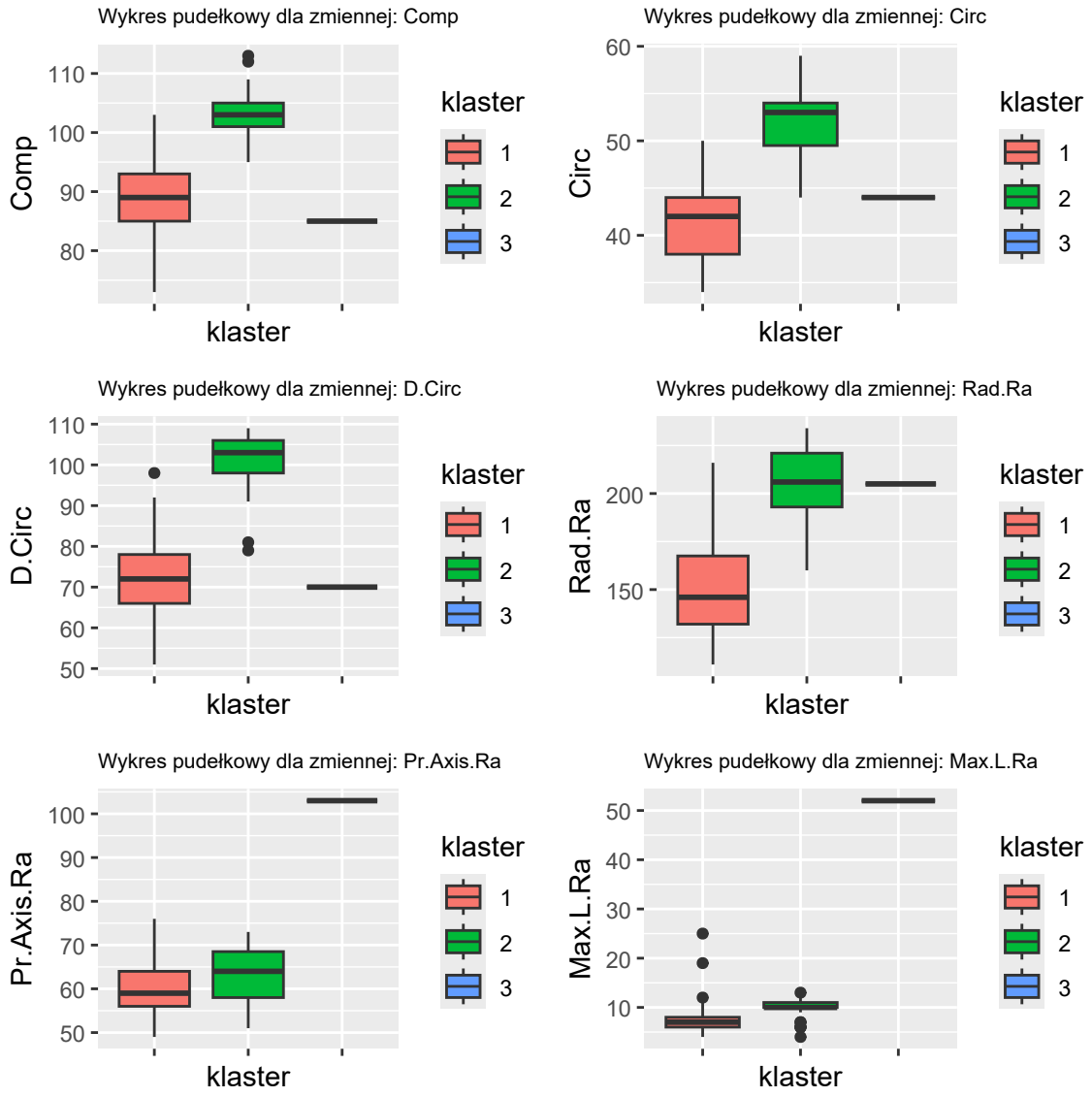
(ciężko sprawdzić zwartość i separalność jednoelementowego klastra). Bez tych zer, te średnie to odpowiednio 0.4442569 i 0.391176.

Co do zgodności klas, to metoda `average` ma skuteczność na poziomie 38.5%, `complete` ma 37.5%, a `single` tylko 28. Skuteczności są podobne do przypadku, gdy k = 4.

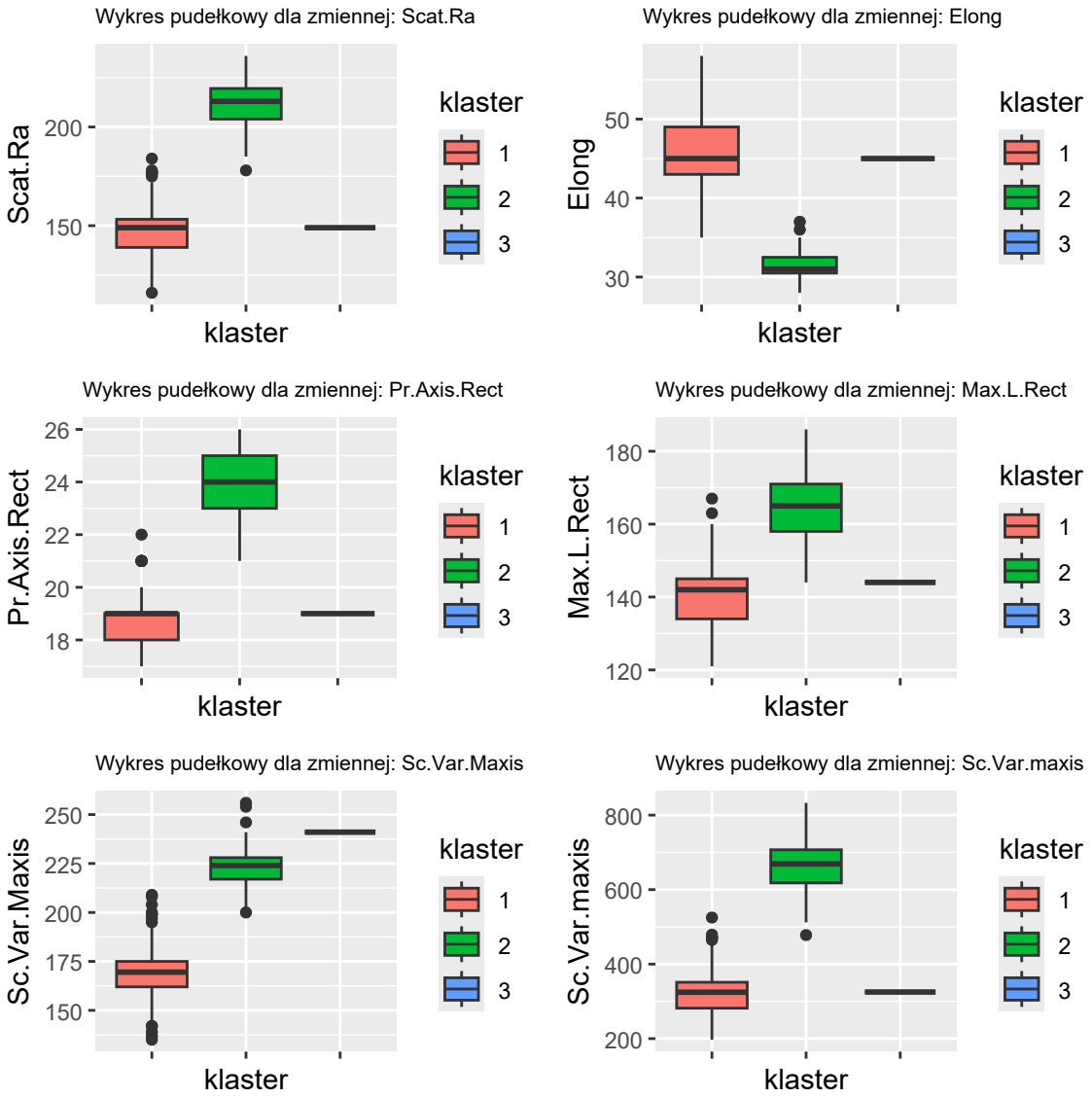
Biorąc pod uwagę wszystkie kombinacje, w algorytmie AGNES **najlepiej prezentuje się metoda `average` dla k = 3**, która ma 38.5% skuteczności i indeks silhouette na rzeczywistym poziomie 0.4442569.

Tabela 10: Średnie wartości cech obiektów z danych klastrów, algorytm AGNES, metoda average

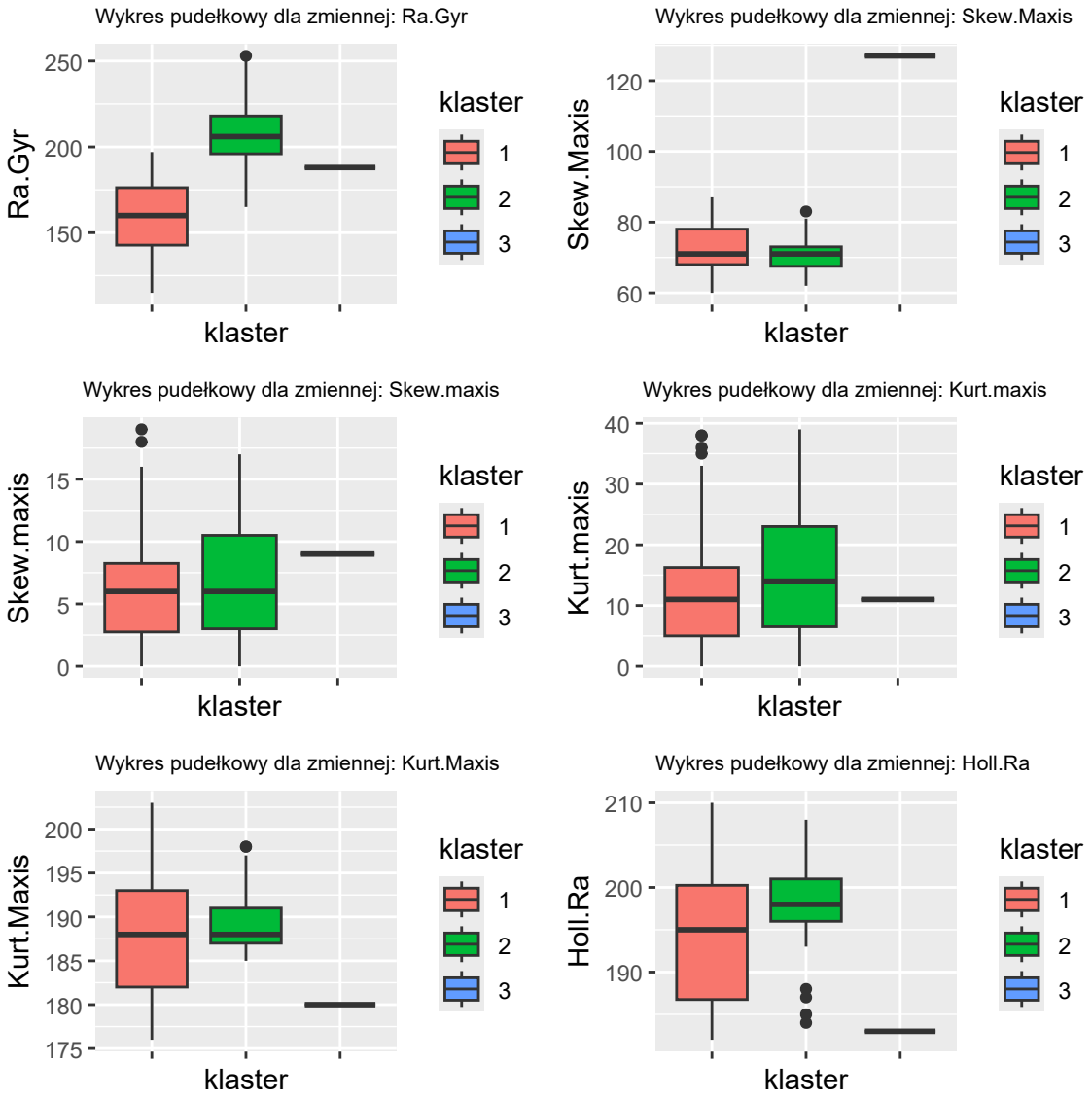
	1	2	3
Comp	88.507143	102.779661	85
Circ	41.435714	51.983051	44
D.Circ	72.371429	101.440678	70
Rad.Ra	150.300000	204.254237	205
Pr.Axis.Ra	60.621429	63.508475	103
Max.L.Ra	7.550000	10.016949	52
Scat.Ra	147.721429	209.864407	149
Elong	45.642857	31.711864	45
Pr.Axis.Rect	18.921429	23.779661	19
Max.L.Rect	140.207143	164.016949	144
Sc.Var.Maxis	169.485714	223.542373	241
Sc.Var.maxis	326.442857	656.033898	325
Ra.Gyr	158.428571	207.118644	188
Skew.Maxis	73.028571	70.762712	127
Skew.maxis	6.007143	6.830508	9
Kurt.maxis	12.121429	15.440678	11
Kurt.Maxis	188.035714	189.220339	180
Holl.Ra	193.871429	198.118644	183



Rysunek 16: Wykresy pudełkowe zmiennych z podziałem na klastry, algorytm AGNES, metoda average



Rysunek 17: Wykresy pudełkowe zmiennych z podziałem na klastry, algorytm AGNES, metoda average



Rysunek 18: Wykresy pudełkowe zmiennych z podziałem na klastry, algorytm AGNES, metoda average

Tabela 10 i rysunki 16-18 pokazują, że wiele cech (np.: `Scat.Ra`, `Elong`, `Comp`, `D.Circ`) pozwala na dobre rozróżnienie pierwszego i drugiego klastra. Poziomie kreski w przypadku trzeciej grupy wynikają oczywiście z tego, że zawiera ona jeden obiekt. Pierwsze skupienie na pewno wyróżnia się niskimi wartościami w zmiennych `Sc.Var.Maxis`, `Pr.Axis.Rect` czy też `Rad.Ra`. Z kolei drugi klaster ma małe wartości dla cechy `Elong`, a w pozostałych przypadkach ma wartości podobne lub większe od pierwszej grupy. Trzeci klaster ma zazwyczaj podobne wartości jak pozostałe dwa, poza takimi przypadkami jak w `Pr.Axis.Ra`,`Max.L.Ra` czy chociażby `Skew.Maxis`.