Sprawozdanie z listy nr 4

Eksploracja Danych

Mateusz Cieślak, Dawid Skowroński

2025-06-26

Spis treści

Spis wykresów

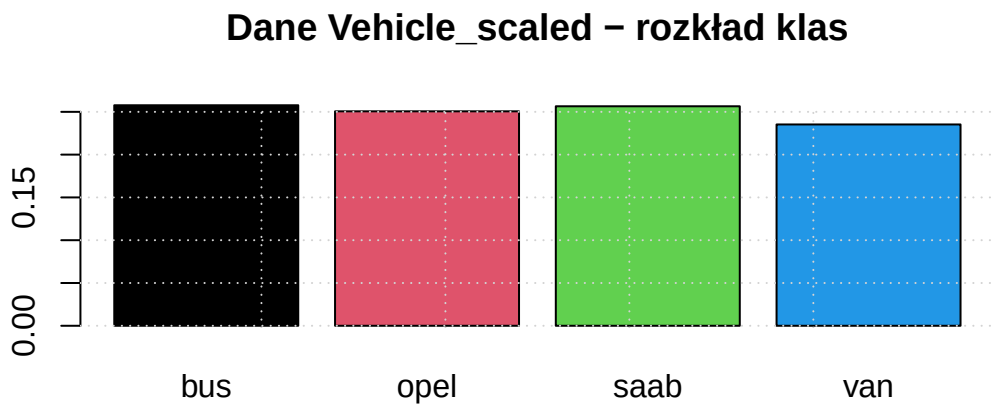
Spis tabel

1 Zadanie 1

1.1 Ensemble learning

W tej sekcji zastosujemy zaawansowane metody klasyfikacji – tzw. metody **ensemble learning**, takie jak **bagging**, **boosting** i **random forest**. Jest to kontynuacja analizy zbioru danych `Vehicle`, który wybraliśmy w poprzednim sprawozdaniu.

1.1.1 Rozkład klas



Wykres 1: Wykres słupkowy przedstawiający liczbę obserwacji dla poszczególnej klasy

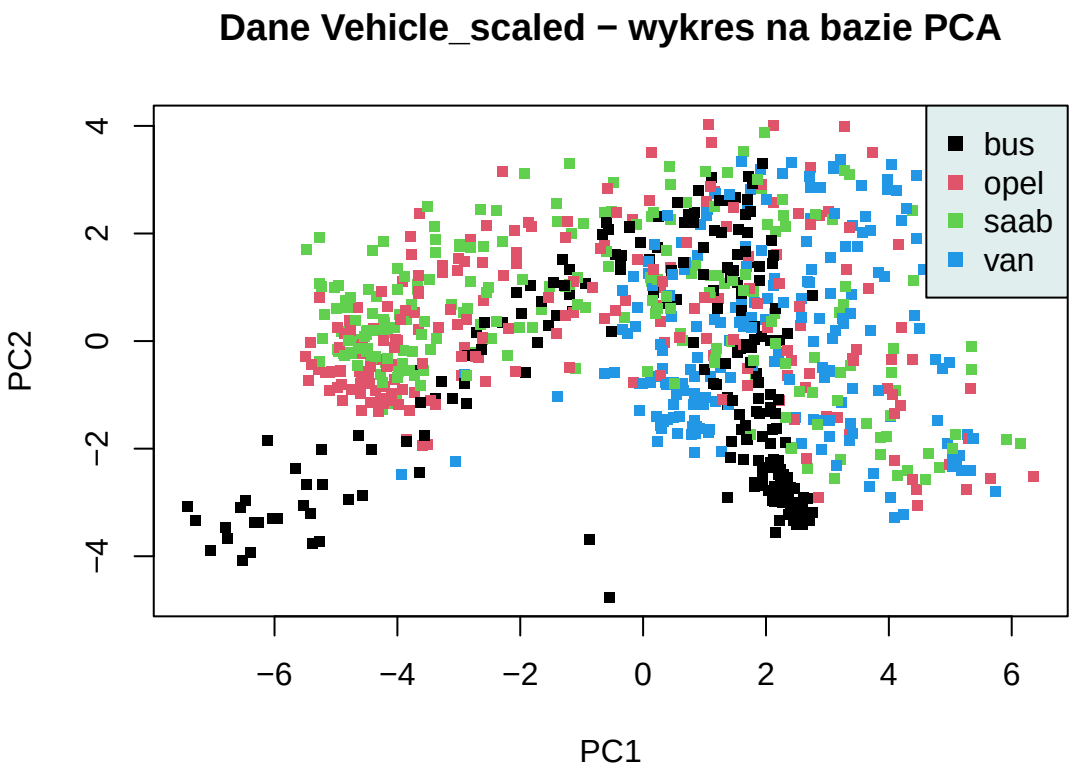
Przypominamy, że klasy w zbiorze Vehicle są równomiernie rozłożone (wykres 1) i nie zawiera on wartości NA (suma NA = 0). Ponadto jest w nim jedna zmienna typu factor "Class", która zawiera informacje o przynależności do jednej spośród czterech klas: bus, opel, saab, van. Pozostałe 18 to zmienne numeryczne.

1.1.2 PCA w 2D czyli wstępna eksploracja danych

Aby uzyskać intuicyjne wyobrażenie o strukturze danych i separowalności klas, przeprowadzamy analizę głównych składowych (PCA).

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.0705 1.7386 1.3779 1.08728 0.95400 0.73033 0.59669
## Proportion of Variance 0.5238 0.1679 0.1055 0.06568 0.05056 0.02963 0.01978
## Cumulative Proportion  0.5238 0.6917 0.7972 0.86287 0.91344 0.94307 0.96285
##                           PC8     PC9    PC10   PC11    PC12    PC13    PC14
## Standard deviation     0.46989 0.39755 0.30218 0.2508 0.20940 0.18721 0.14610
## Proportion of Variance 0.01227 0.00878 0.00507 0.0035 0.00244 0.00195 0.00119
## Cumulative Proportion  0.97511 0.98389 0.98897 0.9925 0.99490 0.99685 0.99803
##                           PC15    PC16    PC17    PC18
## Standard deviation     0.12646 0.11355 0.07870 0.01904
## Proportion of Variance 0.00089 0.00072 0.00034 0.00002
## Cumulative Proportion  0.99892 0.99964 0.99998 1.00000
```

Pierwsze dwie składowe wyjaśniają 69% wariancji, więc dane można wstępnie dobrze zobrazować na płaszczyźnie 2D.

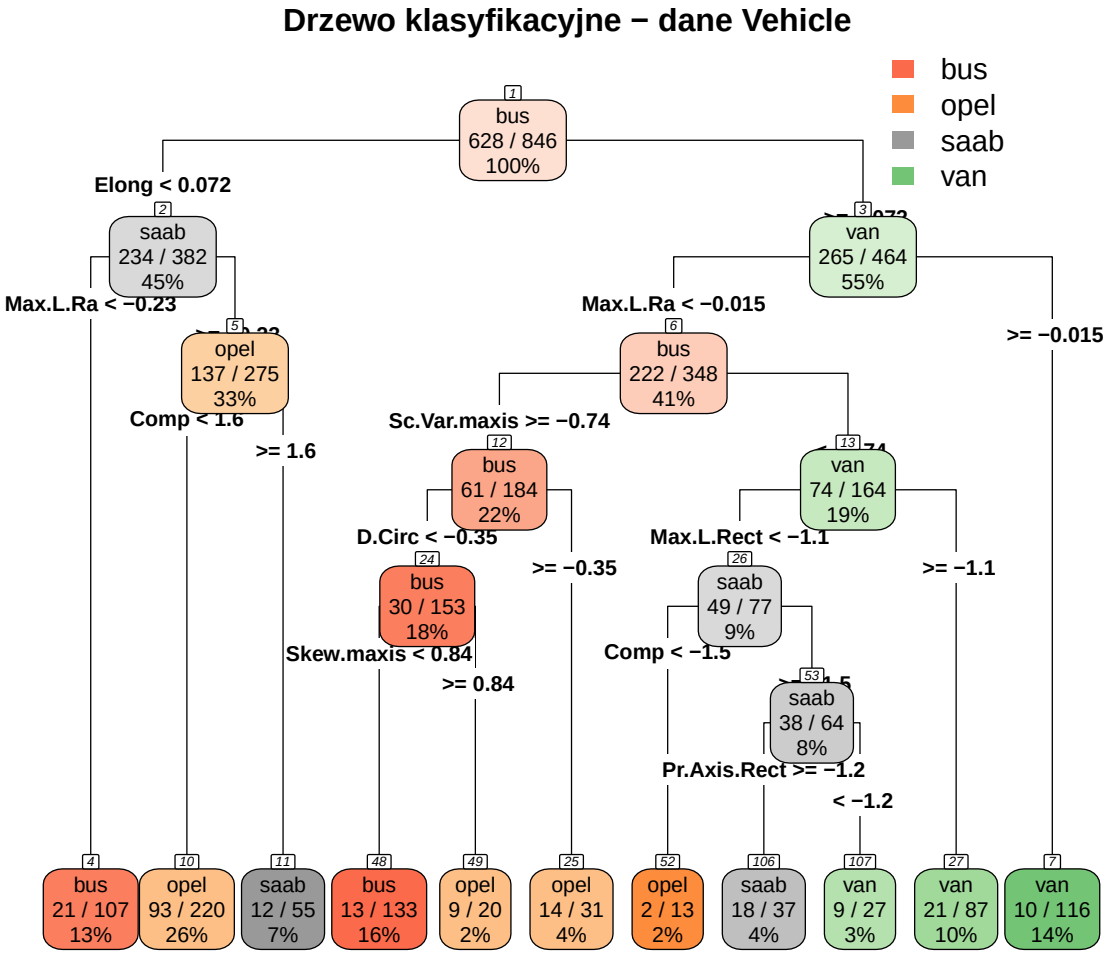


Wykres 2: Wstępna analiza odseparowania klas z użyciem PCA 2D

Możemy zauważyć, że nawet po wykonaniu PCA (wykres 2) klasy są bardzo słabo odseparowane, może to sugerować, że klasyfikacja będzie trudna.

1.1.3 Pojedyncze drzewo klasyfikacyjne

Na początek zastosujemy klasyczny model pojedynczego drzewa decyzyjnego jako klasyfikator bazowy, który posłuży jako punkt odniesienia przy ocenie skuteczności bardziej zaawansowanych metod ensemble learning.

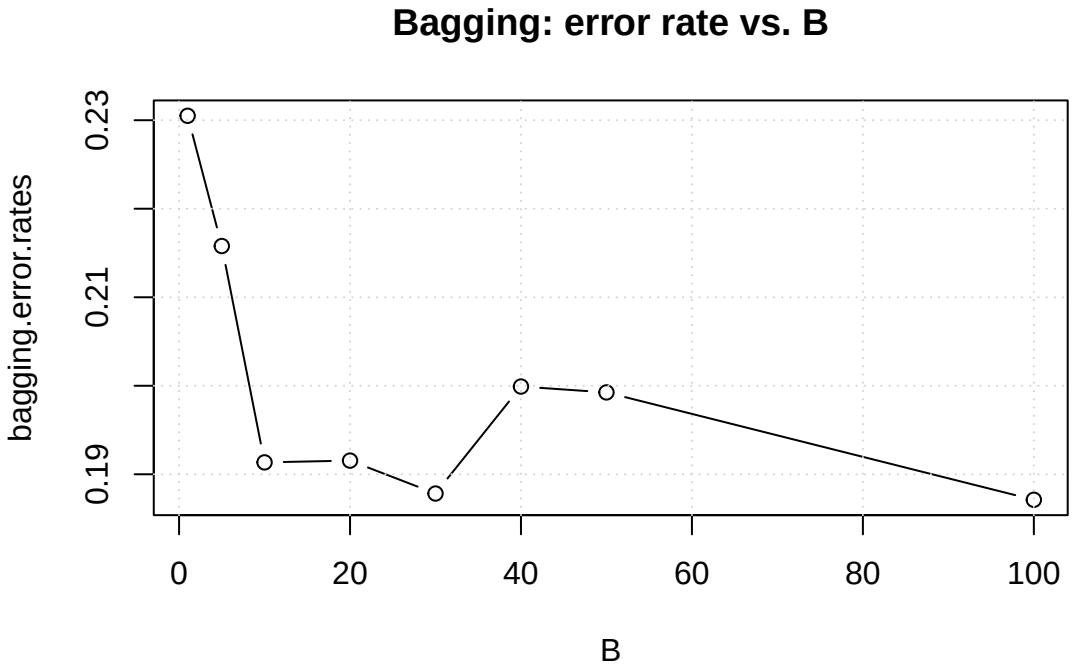


Wykres 3: Pojedyncze drzewo klasyfikacyjne

Błąd dla pojedynczego drzewa klasyfikacyjnego (obliczony metodą 632plus) wynosi 29.9%.

1.1.4 Algorytm bagging

Bagging (skrót od Bootstrap Aggregating) to metoda zespołowa polegająca na tworzeniu wielu klasyfikatorów bazowych (np. drzew decyzyjnych) uczonych na losowanych ze zwracaniem podzbiorach danych treningowych. Ostateczna predykcja powstaje poprzez głosowanie większościowe (dla klasyfikacji). Celem tej sekcji jest zbadanie, jak liczba drzew wpływa na błąd klasyfikacji, oraz porównanie skuteczności baggingu z klasyfikacją opartą na pojedynczym drzewie.



Wykres 4: Wykres przedstawiający zależność błędu od liczby replikacji B

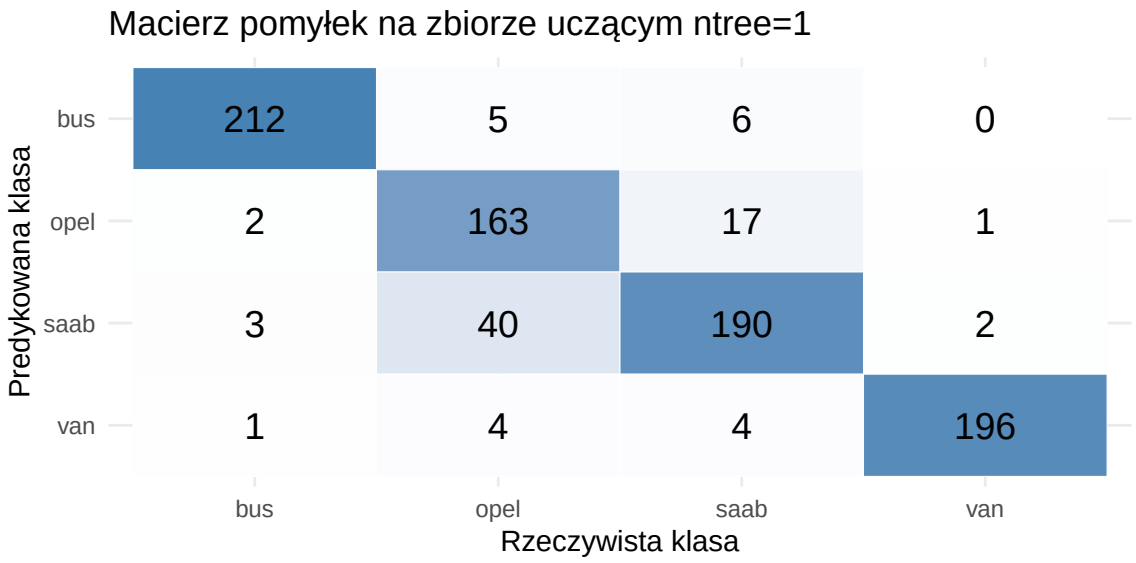
Wykres 4 pozwala zauważyć, że dla około 20 replikacji błąd znacząco się zmniejsza.

Błąd klasyfikacji dla B = 20 uzyskany metodą 632plus wynosi 19.1%.

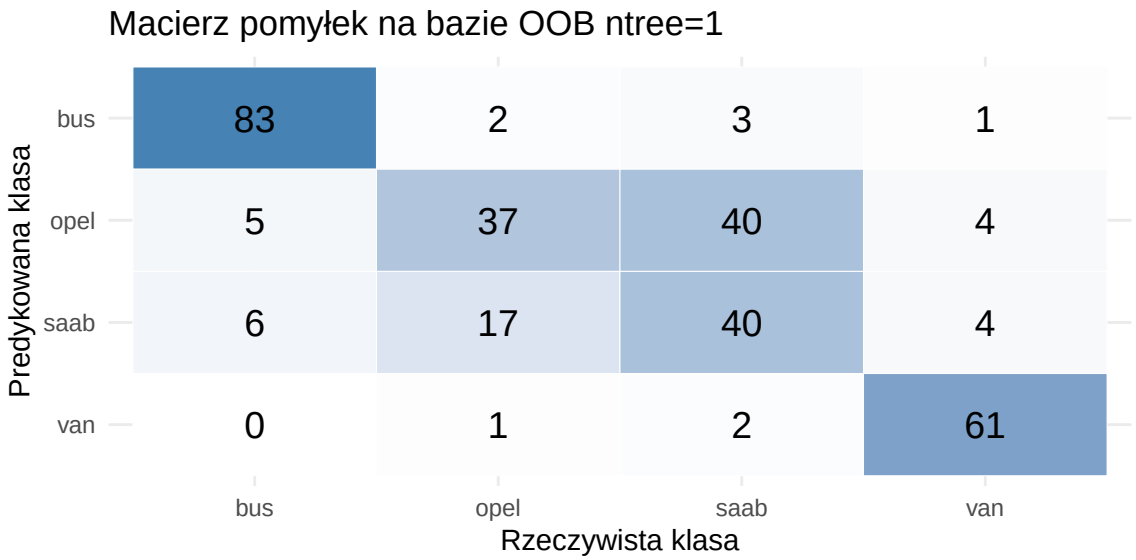
1.1.5 Random forest

Random Forest (las losowy) to jedna z najskuteczniejszych metod klasyfikacji opartych na drzewach decyzyjnych. Polega na budowie wielu drzew decyzyjnych na losowych próbkach danych oraz losowych podzbiorach zmiennych przy każdym podziale. Predykcja końcowa opiera się na głosowaniu większościowym (dla klasyfikacji) lub uśrednianiu (dla regresji). Dzięki losowości i agregacji wielu modeli, Random Forest jest metodą odporną na przeuczenie i charakteryzuje się wysoką skutecznością, nawet bez specjalnego dostrajania hiperparametrów.

Najpierw zaczniemy od sprawdzenia błędu dla parametru `ntree = 1`.



Wykres 5: Macierz pomyłek dla parametru ntree = 1

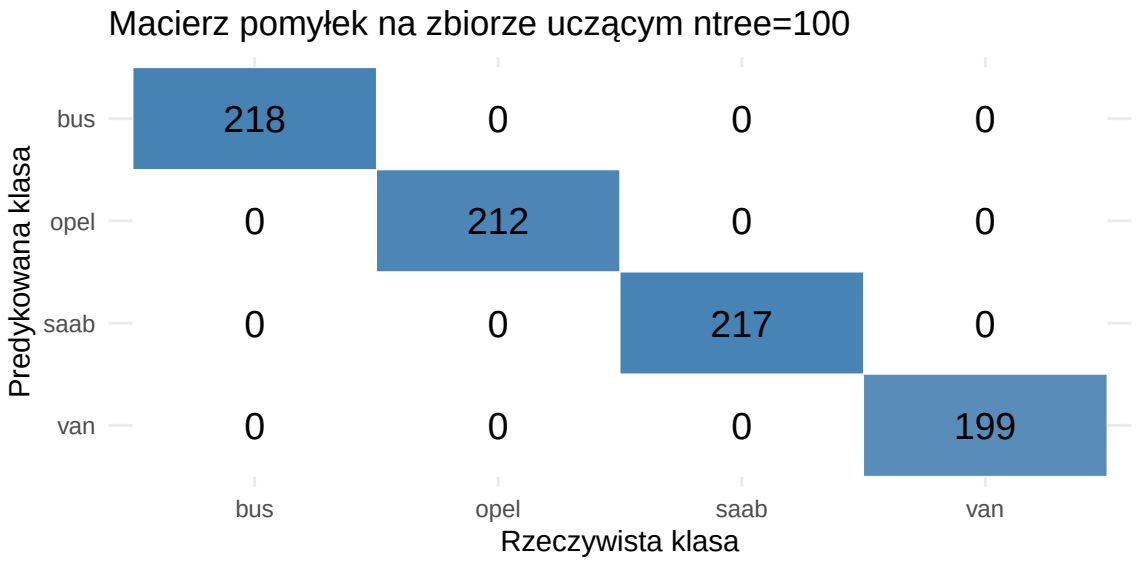


Wykres 6: Macierz pomyłek dla parametru ntree = 1 na bazie OOB

Szacowany błąd na bazie out of bag: 27.78%.

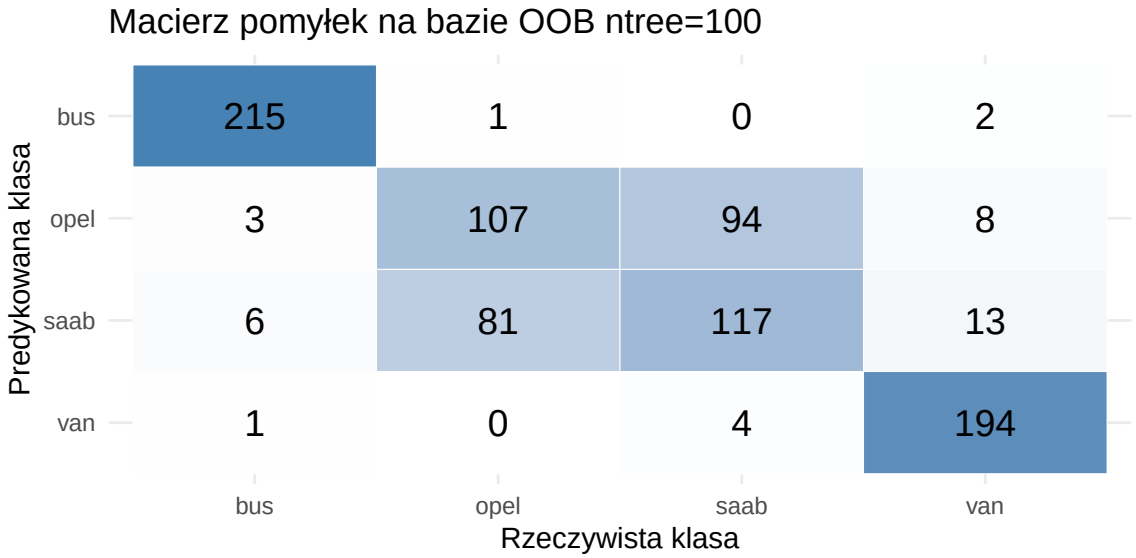
Różnica nie jest znacząca w porównaniu z pojedynczym drzewem klasyfikacyjnym, ponieważ w zasadzie tutaj również testujemy pojedyncze drzewo.

Sprawdźmy teraz czy wybierając `ntree = 100` poprawimy wynik.



Wykres 7: Macierz pomyłek dla parametru ntree = 100

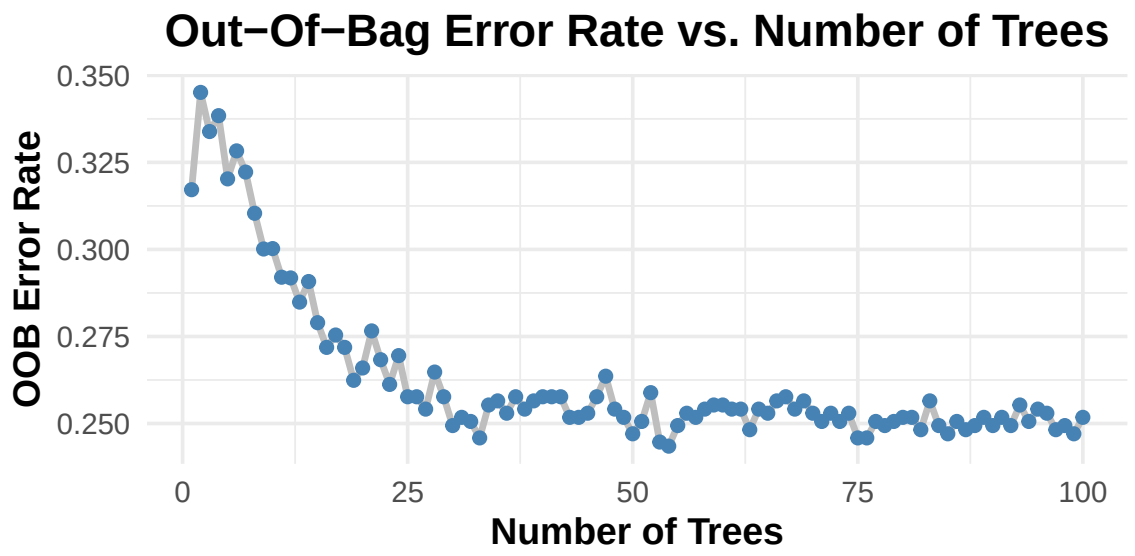
Jak widać na wykresie 7 model perfekcyjnie dopasował się do zbioru uczącego, ale to oznacza przeuczenie.



Wykres 8: Macierz pomyłek dla parametru ntree = 100 na bazie OOB

Szacowany błąd na bazie out of bag: 25.18 %.

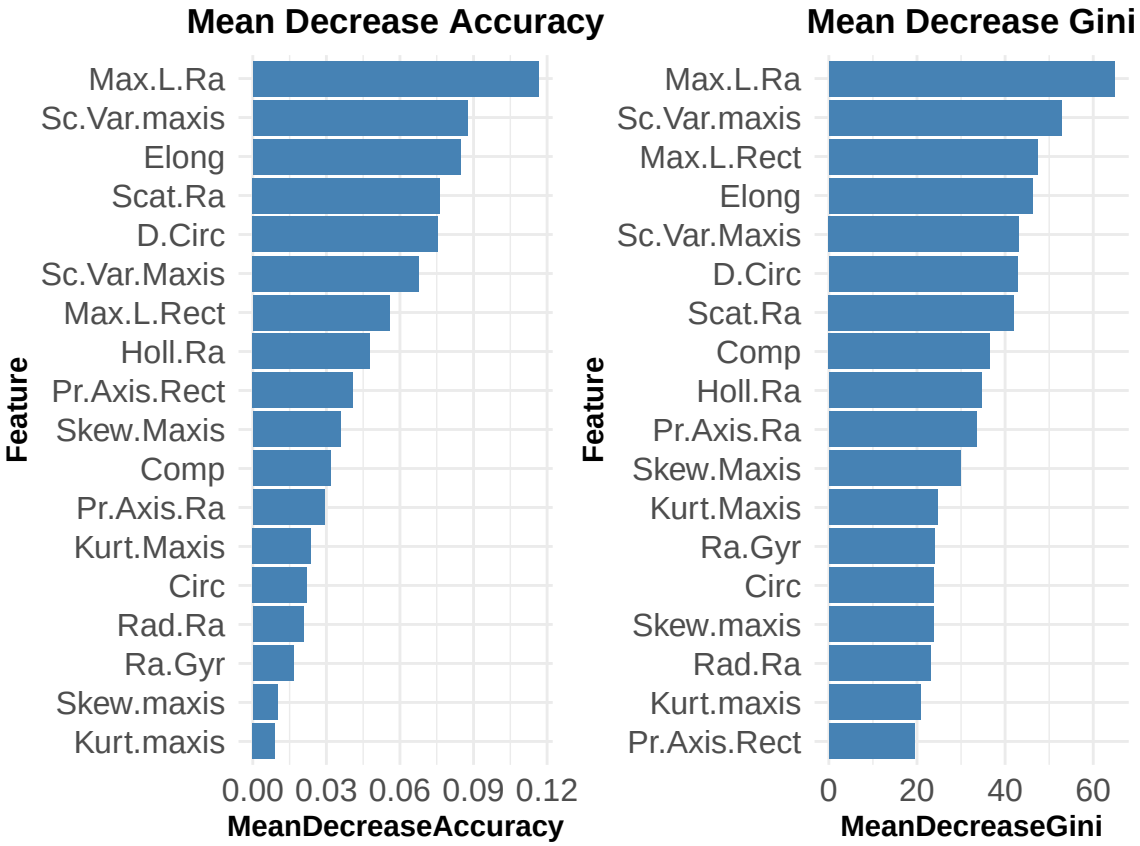
Różnica dla `ntree = 100` wynosi około 2.6 punktów procentowych względem `ntree = 1`.



Wykres 9: Wykres przedstawiający błąd klasyfikacji na podstawie OOB w zależności od liczby drzew

Na podstawie wykresu 9 widzimy, że już dla około 30-50 drzew błąd jest bardzo niski i większa liczba drzew niekoniecznie go poprawia.

Spójrzmy, które cechy są kluczowe dla naszego modelu.



Wykres 10: Ważność cech dla modelu ntree=100

Zauważamy, że `Max.L.Ra`, `Sc.Var.maxis` i `Elong` to najważniejsze cechy w tym modelu random forest. Wracając do wykresu 3 możemy dodatkowo zauważyć, że dla pojedynczego drzewa ważne były podobne cechy.

Spójrzmy ile wynosi błąd testowany metodą 632plus, ponieważ może się on różnić od błędów na bazie out of bag.

Błąd dla metody random forest (`n=100`) sprawdzany metodą 632plus: 18.17%.

Jest on jeszcze niższy niż błąd na bazie out of bag.

1.1.6 Podsumowanie dokładności metod (metoda .632+)

- Błąd dla pojedynczego drzewa: 29.88%
- Błąd dla metody bagging: 19.14%
- Błąd dla metody random forest: 18.17%

Względna redukcja błędu klasyfikacji (w %) w porównaniu z pojedynczym drzewem

bagging vs. single tree: 35.94
random forest vs. single tree: 39.18

Przypomnijmy błędy uzyskane prostszymi metodami w poprzednim raporcie:

- Metoda `k-NN`: około 25%
- Drzewo klasyfikacyjne: około 32%
- Naive Bayes: około 54 %

Wnioski

- W poprzednim raporcie najskuteczniejsza okazała się metoda `k-NN`, której błąd wynosił około 25%, jednakże używając bardziej zaawansowanych metod takich jak random forest zmniejszyliśmy go do zaledwie 18.17% co jest znacznie niższym wynikiem.
- Błąd uzyskany metodą random forest jest również znacznie niższy od błędu na pojedynczym drzewie.
- Różnica pomiędzy różnymi metodami konstrukcji klasyfikatorów złożonych nie jest już tak duża, ale zazwyczaj random forest daje mniejszy błąd niż bagging.

1.2 Metoda wektorów nośnych (SVM)

Support Vector Machines (SVM) to jedna z najskuteczniejszych metod klasyfikacji, szczególnie dobrze sprawdzająca się w zadaniach z danymi o wysokim wymiarze. Kluczową rolę w działaniu SVM odgrywa funkcja jądrowa (kernel), która pozwala modelowi odwzorować dane do przestrzeni wyższych wymiarów, umożliwiając liniową separację nawet nieliniowych zbiorów danych. W niniejszym zadaniu zbadamy wpływ rodzaju funkcji jądrowej (liniowej, wielomianowej i radialnej) oraz parametru kosztu C na skuteczność klasyfikatora.

1.2.1 Jądro liniowe dla różnych wartości parametru kosztu

Sprawdzimy jak zachowuje się procent predykcji poprawnie dokonanych przez model i jak zachowuje się błąd wyliczony metodą 632plus przy różnym parametrze kosztu `c`.

Tabela 1: Porównanie metryk dla SVM z kernelem liniowym przy różnych wartościach c

Metryka	c = 0.1	c = 1	c = 10
Accuracy [%]	76.95	81.91	82.27
Błąd .632+ [%]	20.82	20.82	20.96

Accuracy zwiększyło się dla parametru `c = 1` i `c = 10` względem `c = 0.1`, natomiast błąd obliczany metodą 632plus jest podobny dla wszystkich `c`.

1.2.2 Porównanie wyników dla różnych stopni jąder wielomianowych

Tabela 2: Porównanie metryk SVM z kernelem wielomianowym (różne stopnie)

Metryka	stopień = 2	stopień = 4
Accuracy [%]	60.99	57.45
Błąd .632+ [%]	20.60	20.67

Zwiększenie stopnia wielomianu pogorszyło accuracy, ale błąd uzyskany metodą 632plus wciąż jest podobny.

1.2.3 Porównanie wyników dla różnych parametrów gamma jądra radialnego

Tabela 3: Porównanie metryk SVM z kernelem radialnym (różne gamma)

Metryka	gamma domyślna	gamma = 0.1	gamma = 1
Accuracy [%]	71.99	71.63	64.89
Błąd .632+ [%]	20.64	20.73	20.75

Najwyższe accuracy otrzymujemy dla domyślnej gammy.

Podsumowanie

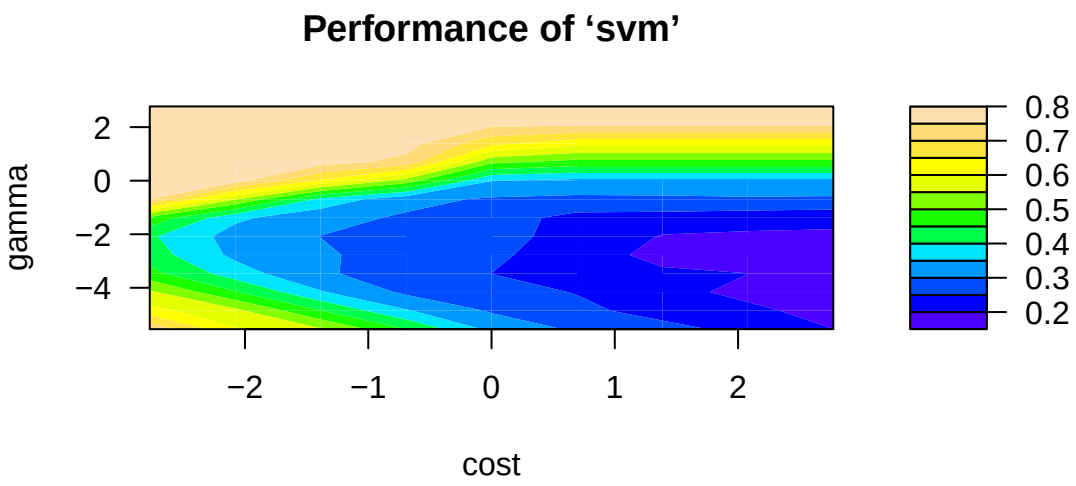
- Wyniki otrzymane za pomocą sprawdzenia procentu poprawnych predykcji (accuracy) są najlepsze dla jądra liniowego, ale błędy klasyfikacji uzyskane metodą 632plus są podobne we wszystkich modelach.
- Na accuracy najbardziej wpływa wybór funkcji jądrowej, ale dobranie odpowiedniego parametru takiego jak cost, stopień wielomianu i gamma mogą ją zauważalnie poprawić lub pogorszyć.

1.2.4 Dostrajanie parametrów

Dobierzemy teraz optymalne parametry `c` i `gamma` dla jądra radialnego, tak model był jak najbardziej skuteczny.

Najlepsze parametry:

- cost (c) = 16
- gamma = 0.03125



Wykres 11: Wykres przedstawiający accuracy SMV z jądrem radialnym w zależności od gammy i costu

Tabela 4: Porównanie dokładności i błędu .632+ dla modelu SVM radialnego (transponowane)

Metryka	SVM radial domyślne	SVM radial optymalne
Accuracy [%]	71.99	80.85
Błąd .632+ [%]	20.64	15.72

Optymalnie dobrane parametry poprawiły nie tylko accuracy, ale również błąd obliczany metodą 632plus co jest zaskakujące, ponieważ w poprzednich testach wybór jądra i parametrów nie wpływał znacznie na błąd obliczany tą metodą, a teraz udało się go znacznie zmniejszyć.

1.3 Porównanie skuteczności metod

W algorytmach ensemble learning najskuteczniejszy okazał się random forest (18.17%). Spośrod metod SVM największe accuracy uzyskał kernel liniowy z `c = 10`, natomiast ta miara może być myląca, ponieważ wynik mógł być przypadkowy z powodu wyboru konkretnego zbioru. Biorąc pod uwagę metodę 632plus w SVM najskuteczniejszy jest klasyfikator z jądrem radialnym, który jest odpowiednio dostrojony. To on jest najskuteczniejszy ze wszystkich metod stosowanych do klasyfikacji w tym jak i poprzednim sprawozdaniu. Jego błąd wyznaczony metodą 632plus wynosi zaledwie 15.72% co jest znaczą poprawą w stosunku do najlepszego `k-NN` z poprzedniego raportu (około 25%) i najlepszego random forest z poprzedniego podpunktu (18.17%). Zatem dla zbioru Vehicle najskutecznejszy okazał się odpowiednio dostrojony SVM z jądrem radialnym.

2 Zadanie 2

W tym zadaniu zastosujemy algorytmy klasteryzacji (grupujące i hierarchiczne) aby odkryć potencjalne ukryte wzorce w danych oraz ich prawdziwą strukturę. Głównym celem jest porównanie metod analizy skupień w kontekście ich skuteczności w grupowaniu danych. Dodatkowo, ocenimy jakość grupowania za pomocą wskaźników wewnętrznych oraz zewnętrznych.

2.1 Wybór i przygotowanie danych

Kontynuujemy pracę ze zbiorem `Vehicle` z pakietu `mlbench`. Zbiór ten zawiera 846 obserwacji, więc w celu zmniejszenia ilości obliczeń i bardziej przejrzystej wizualizacji, wybieramy losowo podzbiór zawierający **200** wierszy.

```r
set.seed(1025)
Vehicle_podzbior <- Vehicle[sample(nrow(Vehicle), size = 200),
    ]
```

Z racji tego, że analiza skupień jest przykładem uczenia nienadzorowanego, musimy usunąć kolumnę zawierającą etykietki klas ze zbioru `Vehicle_podzbior`.

```r
Vehicle_etykiety <- Vehicle_podzbior$Class
Vehicle_dane <- Vehicle_podzbior[, -ncol(Vehicle_podzbior)]
```

2.1.1 Potrzeba standaryzacji

Algorytmy jak **k-means** czy **PAM** są oparte na odległościach między obserwacjami. Dane `Vehicle` zawierają cechy o różnych jednostkach i w różnych skalach. Dlatego standaryzacja jest zalecana, ponieważ spowoduje ona jednakowy wkład zmiennych do metod. Może ona jednak pogorszyć separację klas i wpłynąć negatywnie na skuteczność algorytmów.

```
##         Comp         Circ       D.Circ       Rad.Ra   Pr.Axis.Ra     Max.L.Ra
##        8.590        6.238       15.926       32.711        6.006        2.609
##      Scat.Ra        Elong Pr.Axis.Rect   Max.L.Rect Sc.Var.Maxis Sc.Var.maxis
##       34.305        8.020        2.683       14.248       31.786      183.345
##       Ra.Gyr   Skew.Maxis   Skew.maxis   Kurt.maxis   Kurt.Maxis      Holl.Ra
##       33.257        6.344        4.522        8.184        5.895        7.255
```

Najmniejsza wariancja to **3**, a największa to **183**. Występują zatem dość duże różnice w wariancjach, przez co niektóre zmienne mogą mieć znacznie większy wpływ na klasteryzację, niż pozostałe. Stosujemy zatem standaryzację (w przeciwnym wypadku wpływ cech będzie niezrównoważony).

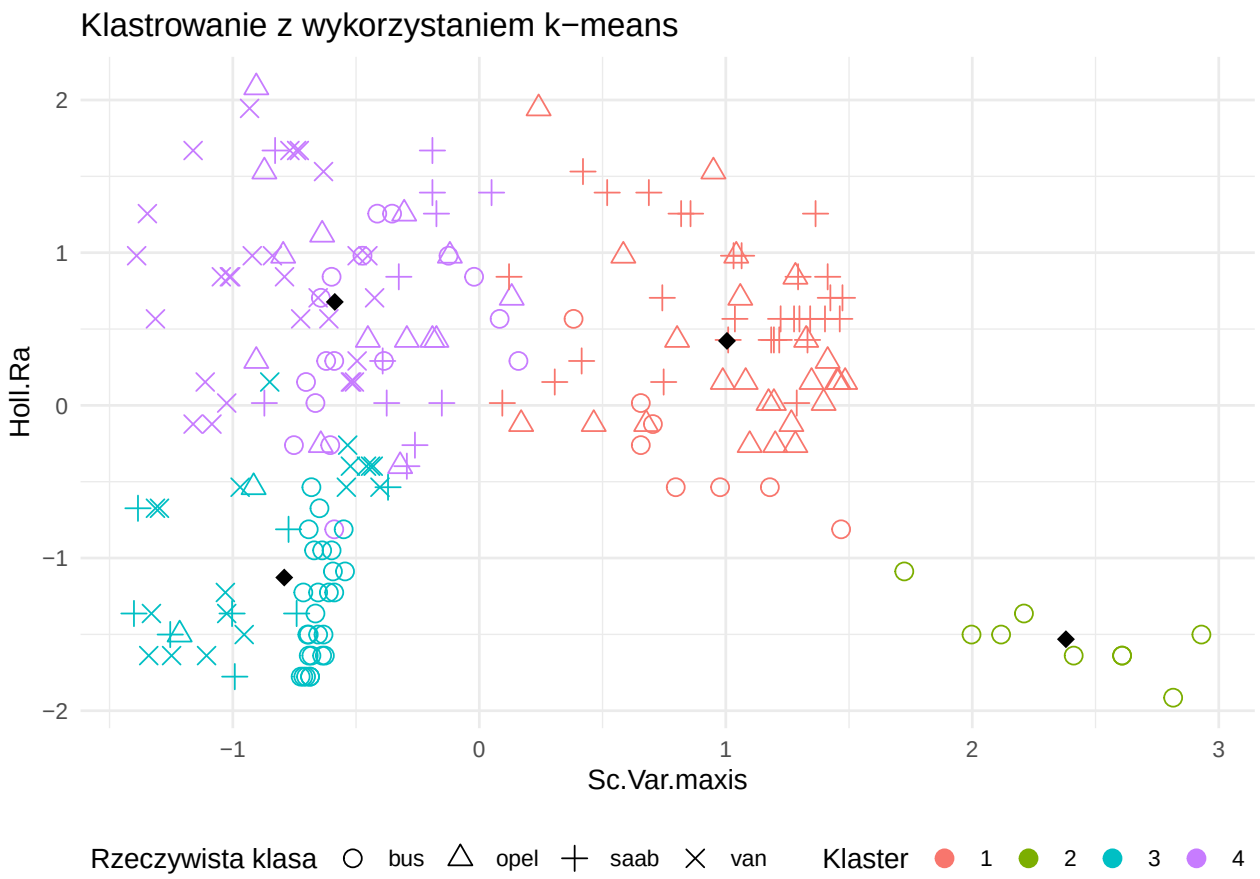
```r
Vehicle_scaled <- scale(Vehicle_dane)
```

2.2 Wizualizacja wyników grupowania

W tej części wykorzystamy różne techniki wizualizacji. Dla metod grupujących zastosujemy wykresy rozrzutu z zaznaczonymi skupieniami, podczas gdy dla metod hierarchicznych przeanalizujemy dendrogramy pokazujące kolejność łączenia obserwacji. Dodatkowo, nałożymy prawdziwe etykiety klas, aby zobaczyć, na ile wyniki grupowania pokrywają się z rzeczywistą strukturą danych. Jako docelową liczbę klastrów przyjmujemy liczbę równą rzeczywistej liczbie typów samochodów w zbiorze `Vehicle` (w tym przypadku dokonujemy podziału na 4 klastry).

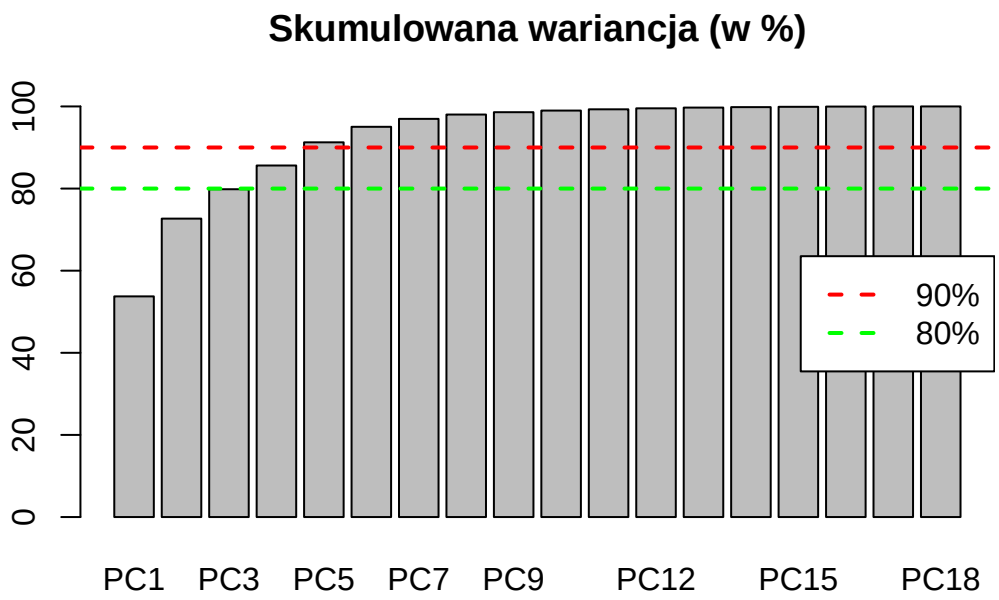
2.2.1 Metody grupujące

Korzystamy z metod takich jak `k-means` oraz `PAM` (Partitioning Around Medoids), które są przykładami metod grupujących, aby wyznaczyć podział danych na 4 klastry.



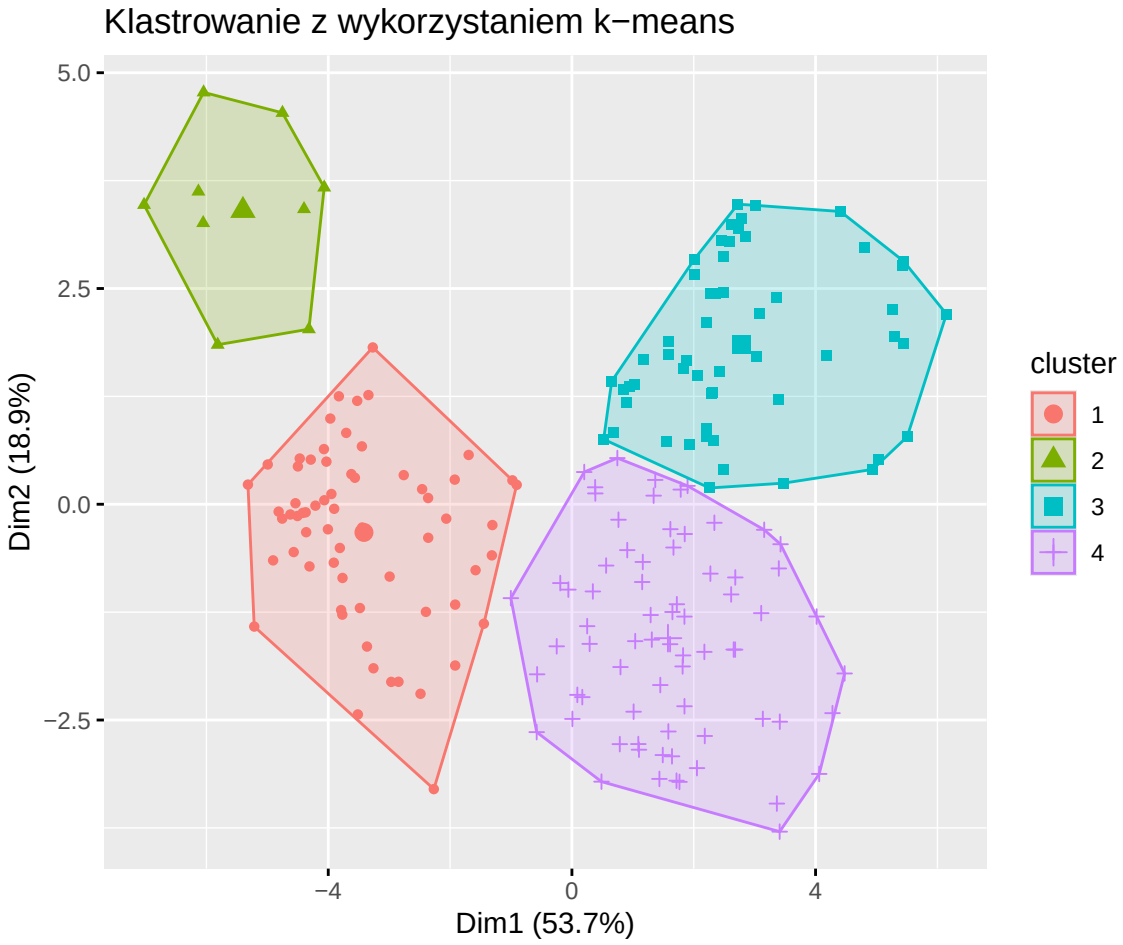
Wykres 12: Wykres rozrzutu zmiennej Holl.Ra od zmiennej Sc.Var.maxis z zaznaczonymi klastrami i centrami klastrów

2.2.1.1 Algorytm k-means Wykres 12 przedstawia zależność między zmiennymi `Holl.Ra` a `Sc.Var.maxis`. Kolorami oznaczone są klastry, wyznaczone za pomocą metody *k-means*, kształtami - rzeczywiste etykiety klas. Centra klastrów oznaczone są za pomocą czarnych rombów. Stworzone klastry są wypukłe (*k-means* z założenia tworzy klastry wypukłe). Możemy zauważyć bardzo dobrą separację klastra **2** (zawiera również jedynie obserwacje o etykiecie **"bus"**), Pozostałe klastry zawierają obserwacje ze wszystkich czterech typów samochodów. Obserwacje klastra **3** tworzą najbardziej zwartą grupę (dodatkowo przeważają w niej obserwacje z klasy **"bus"** i **"van"**). Klasy **"opel"** i **"saab"** silnie się mieszają.



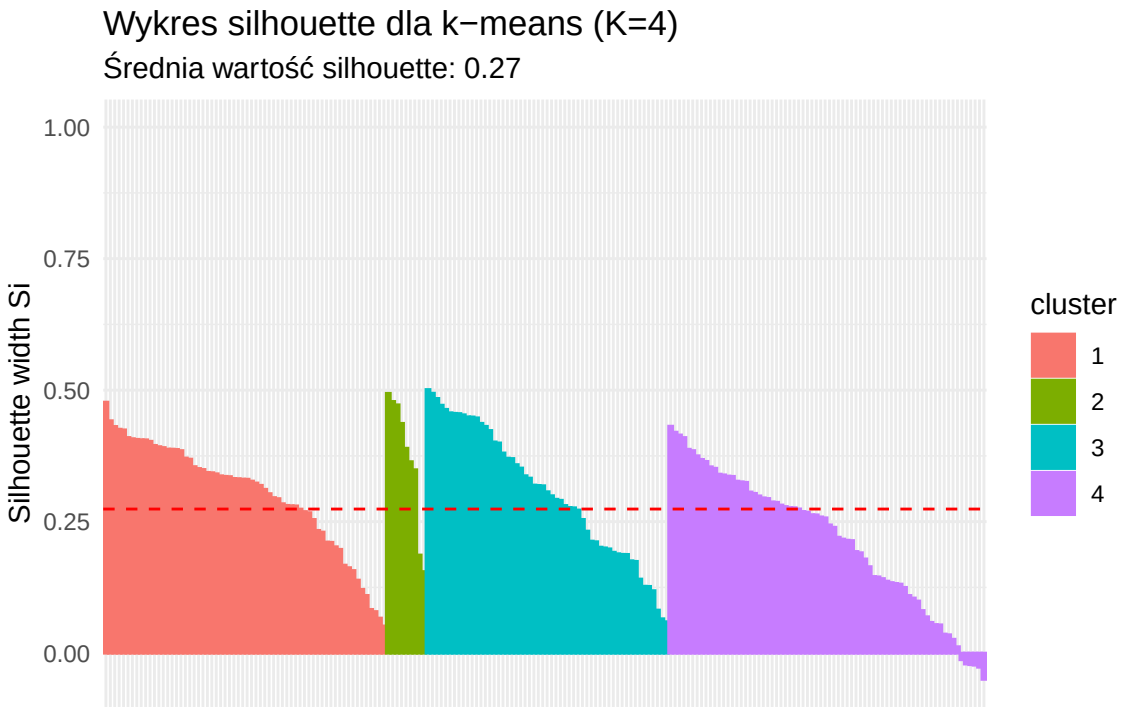
Wykres 13: Skumulowana wariancja PCA

Wykres 13 prezentuje skumulowany procent wariancji wyjaśnionej przez kolejne składowe główne. Pierwsze dwie składowe wyjaśniają około **72.68**%, co może skutkować maskowaniem subtelnych różnic pomiędzy klasami.



Wykres 14: Klastry uzyskane metodą k-means w PCA

Na wykresie 14 można zauważyć wyniki grupowania k-means w przestrzeni dwóch pierwszych składowych głównych. Możemy zauważyć dobrą separację klastra **2** od pozostałych, jednak jego zwartość jest najgorsza spośród wszystkich klastrów. Dodatkowo, klaster ten zawiera zdecydowanie mniej obserwacji niż pozostałe, co świadczy o wrażliwości metody *k-means* na obserwacje odstające. Również separacja przestrzenna pozostałych klastrów jest bardzo niska.



Wykres 15: Silhouette dla k-means

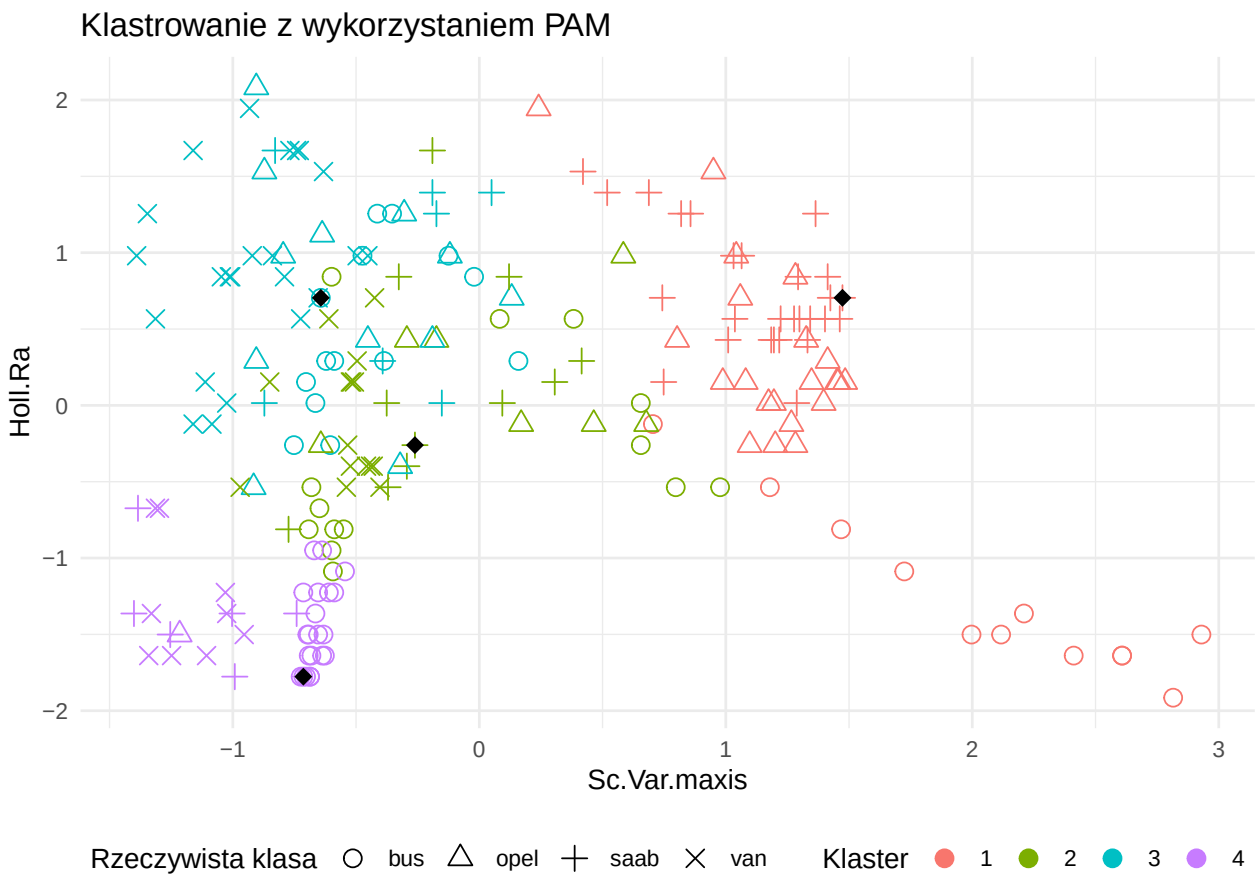
Wykres `silhouette` dla algorytmu *k-means* (15) pokazuje niską średnią wartość wskaźnika (**0.27**). Liczne obserwacje mają niskie lub ujemne wartości `silhouette`, co oznacza, że zostały przypisane do niewłaściwego klastra lub znajdują się blisko granic między klastrami. To sugeruje, że *k-means* może być mniej skuteczny w przypadku danych o nieregularnych strukturach.

Tabela 5: Statystyki silhouette dla każdego klastra (K-means)

Klaster	Średnia wartość silhouette	Liczba obiektów	Liczba ujemnych
1	0.30	64	0
2	0.37	9	0
3	0.30	55	0
4	0.21	72	6

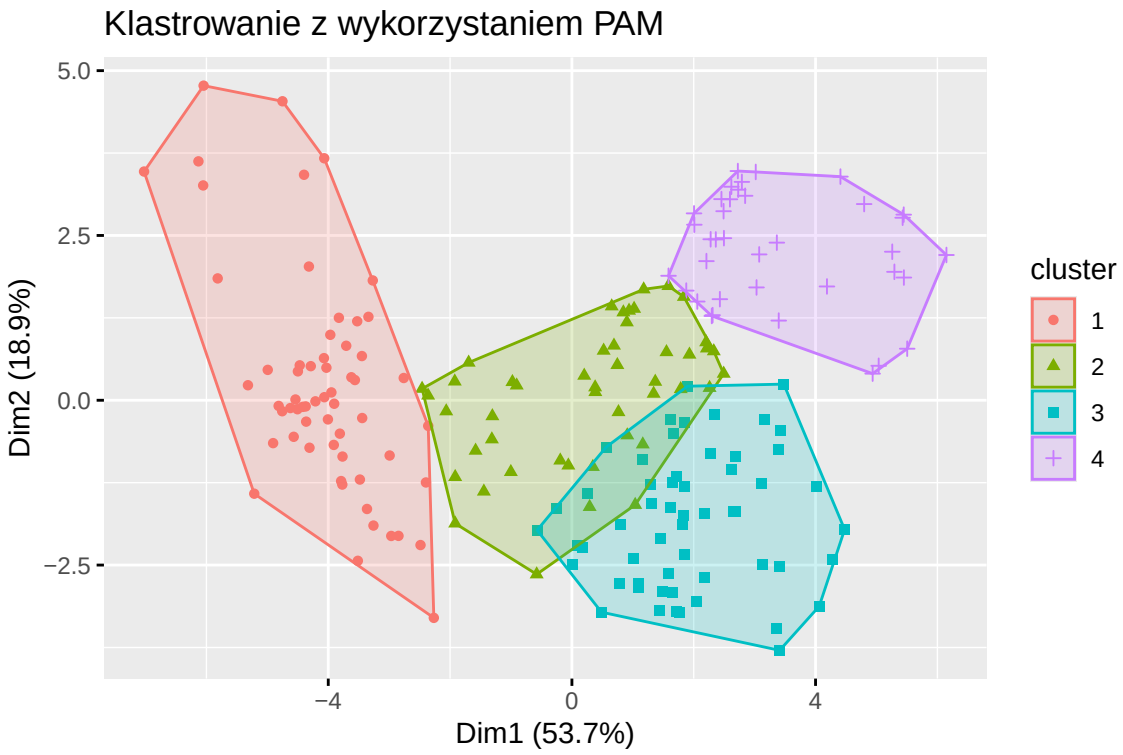
Tabela 5 przedstawia wyniki z wykresu 15. Możemy zauważyć, że największa średnia wartość silhouette została uzyskana dla klastra **2**, jednak zawiera on jedynie 9 obserwacji. Klaster **4** ma natomiast 6 obserwacji z ujemną wartością silhouette – błędnie przypisanych.

Trafność klasyfikacji dla metody k-means po dopasowaniu etykiet wyniosła 43.5%.



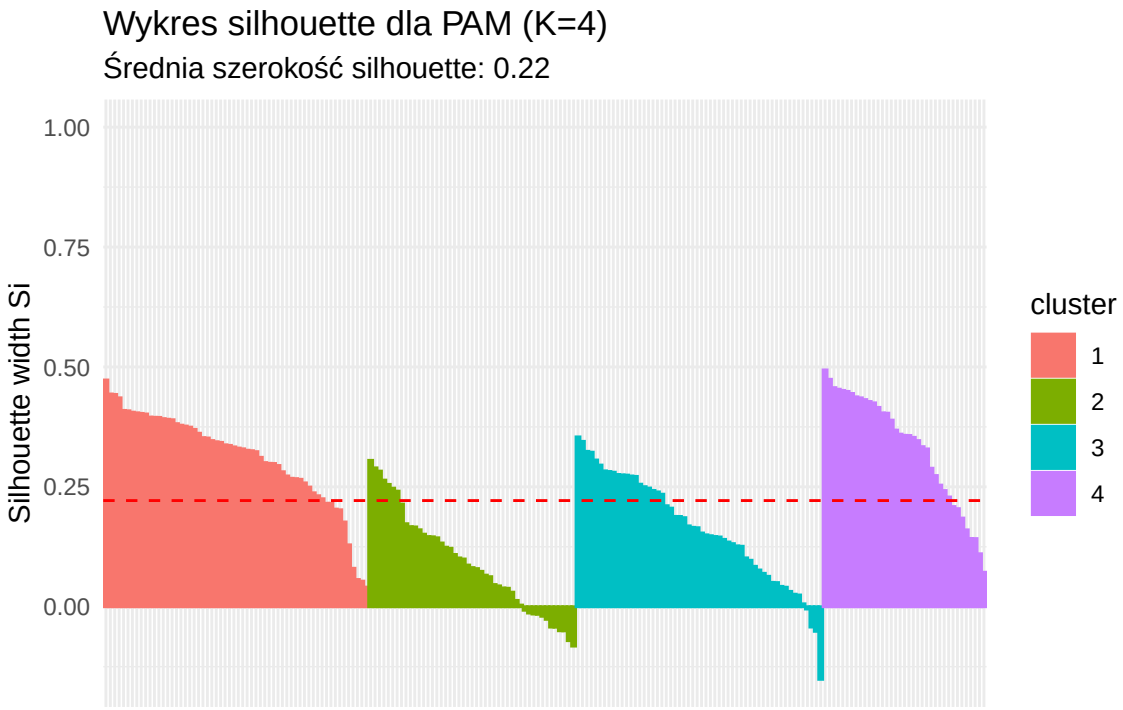
Wykres 16: Wykres rozrzutu zmiennej Holl.Ra od zmiennej Sc.Var.maxis z zaznaczonymi klastrami i medoidami klastrów

2.2.1.2 Algorytm PAM Wykres 16 przedstawia rozrzut obserwacji względem zmiennych `Sc.Var.maxis` i `Holl.Ra`. Medoidy zostały oznaczone poprzez czarne romby. Analizując metodę *PAM* względem metody *k-means*, możemy zauważyć, że klaster **2** z wykresu 12 został przypisany do klastra **1** (wykres `PAM`). Oznacza to że algorytm `PAM` jest bardziej odporny na obserwacje odstające. Dodatkowo, medoidy dla klastrów **1** i **3** znajdują się w obszarach o "gęstej" liczbie obserwacji o etykiecie **"saab"** i **"bus"** odpowiednio. Widoczne jest też znaczne nakładanie się klastrów **2** i **3**.



Wykres 17: Wizualizacja wyników PAM w przestrzeni PCA

Zrzutowanie wyników klasteryzacji metodą PAM (wykres 17) pokazuje dość silne nakładanie się rzutów klastra **2** i **3**. Widoczna jest również o wiele gorsza separacja przestrzenna w porównaniu do metody *k-means* (praktycznie wszystkie rzuty graniczą bezpośrednio z jakimś innym klastrem).



Wykres 18: Wykres wskaźnika silhouette dla metody PAM

Wykres `silhouette` dla algorytmu *PAM* 18 przedstawia niższą średnią wartość tego wskaźnika (**0.22**) niż dla *k-means*. Znaczna większość obserwacji klastrów **1** i **4** ma wartość wskaźnika `silhouette` powyżej średniej wartości. O wiele gorsze wyniki osiągane są dla klastrów **2** i **3**, gdzie jedynie niewielka część obserwacji osiąga dobrą wartość wskaźnika. Dodatkowo, niektóre obserwacje z klastrów **2** i **3** przyjmują ujemną wartość tego wskaźnika, co oznacza, że najprawdopodobniej zostały one przypisane do nieprawidłowego klastra.

Tabela 6: Statystyki silhouette dla każdego klastra (PAM)

Klaster	Średnia wartość silhouette	Liczba obiektów	Liczba ujemnych
1	0.31	60	0
2	0.09	47	12
3	0.16	56	4
4	0.33	37	0

Tabela 6 zawiera podsumowanie wyników wykresu 18. Porównując średnie wartości `silhouette` względem tych, uzyskanych metodą *k-means*, możemy zauważyć pogorszenie wyników dla klastrów **2** i **3** (również zauważalna jest większa liczba nieprawidłowo przypisanych obserwacji). Jedyna poprawa jest widoczna dla klastra **4**.

Trafność klasyfikacji dla metody k-means po dopasowaniu etykiet wyniosła 39%.

2.2.2 Algorytmy hierarchiczne

2.2.2.1 AGNES Korzystamy z metody hierarchicznej `AGNES`, która jest przykładem metody aglomeracyjnej, w celu wyznaczenia podziału danych na 4 klastry. Będziemy również porównywać różne metody łączenia klastrów (*najbliższego sąsiada*, *najdalszego sąsiada* oraz *średniej odległości*).

W celu zastosowania metody `AGNES`, musimy wyznaczyć macierz niepodobieństw zbioru `Vehicle_scaled`.

```r
niepodob_Vehicle <- daisy(Vehicle_scaled)
mac_niepodob_Veh <- as.matrix(niepodob_Vehicle)

# Metody łączenia klastrów
Agnes_avg_Veh <- agnes(x = mac_niepodob_Veh, diss = TRUE,
    method = "average")
Agnes_single_Veh <- agnes(x = mac_niepodob_Veh, diss = TRUE,
    method = "single")
Agnes_complete_Veh <- agnes(x = mac_niepodob_Veh, diss = TRUE,
    method = "complete")
```

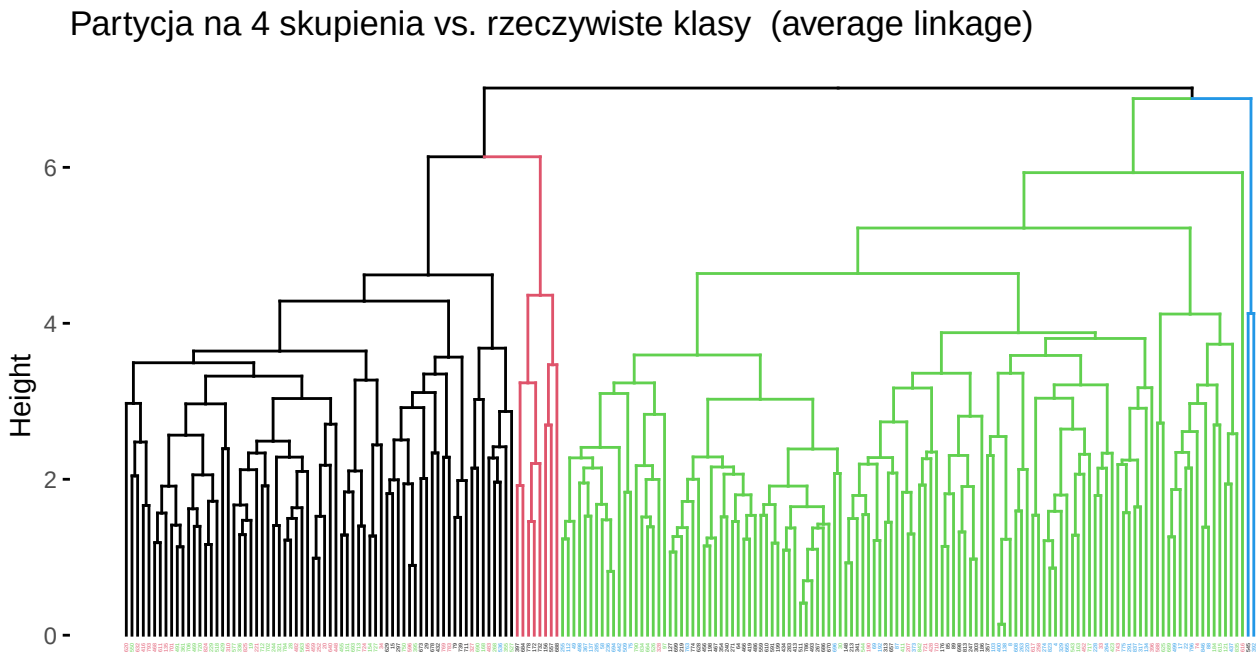
Skuteczności w poniższej tabeli obliczono metodami `matchClasses` i `compareMatchedClasses` - metoda `exact`.

Tabela 7: Porównanie jakości klasteryzacji metodą AGNES (różne metody łączenia)

Metryka	AGNES.average	AGNES.single	AGNES.complete
matchClasses (%)	43.0	31	45.5
compareMatchedClasses (%)	42.5	30	41.5

Najlepsze wyniki osiąga metoda AGNES complete - najlepiej odwzorowuje rzeczywiste etykiety klas, zarówno pod względem poprawnych przypisań, jak i zgodności par obiektów.

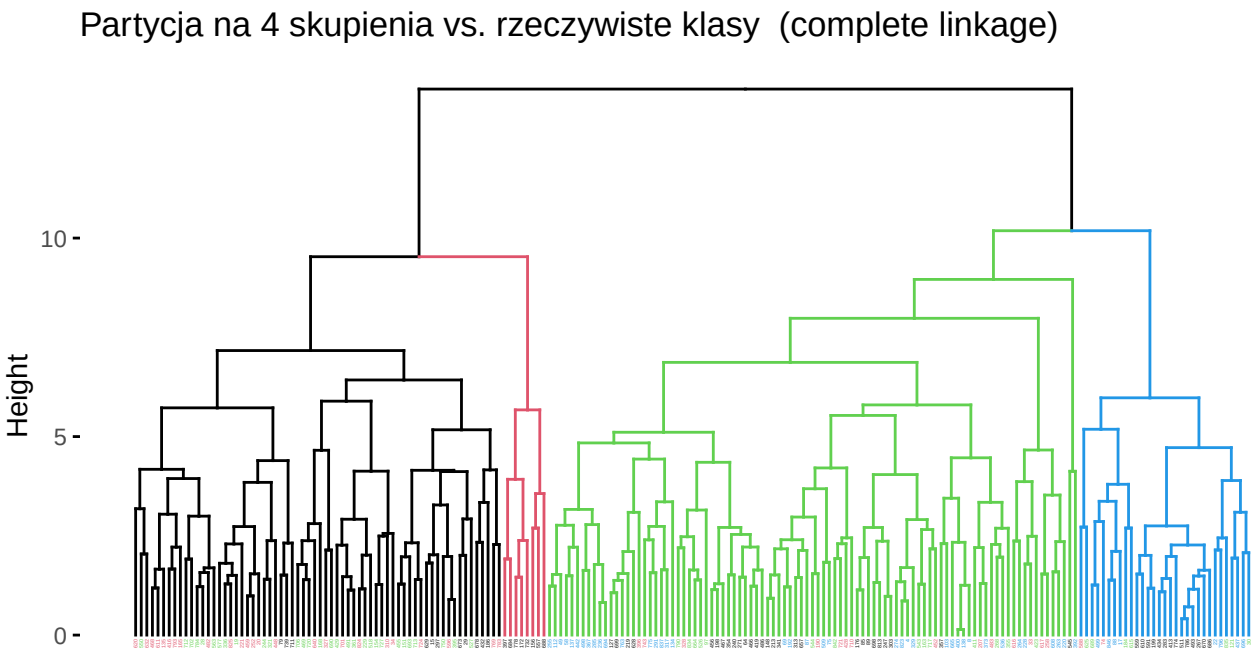
Najsłabiej wypada metoda AGNES single (łączenie najbliższych sąsiadów) słabo radzi sobie z odwzorowaniem struktury klas.



Wykres 19: Dendrogram z użyciem average linkage

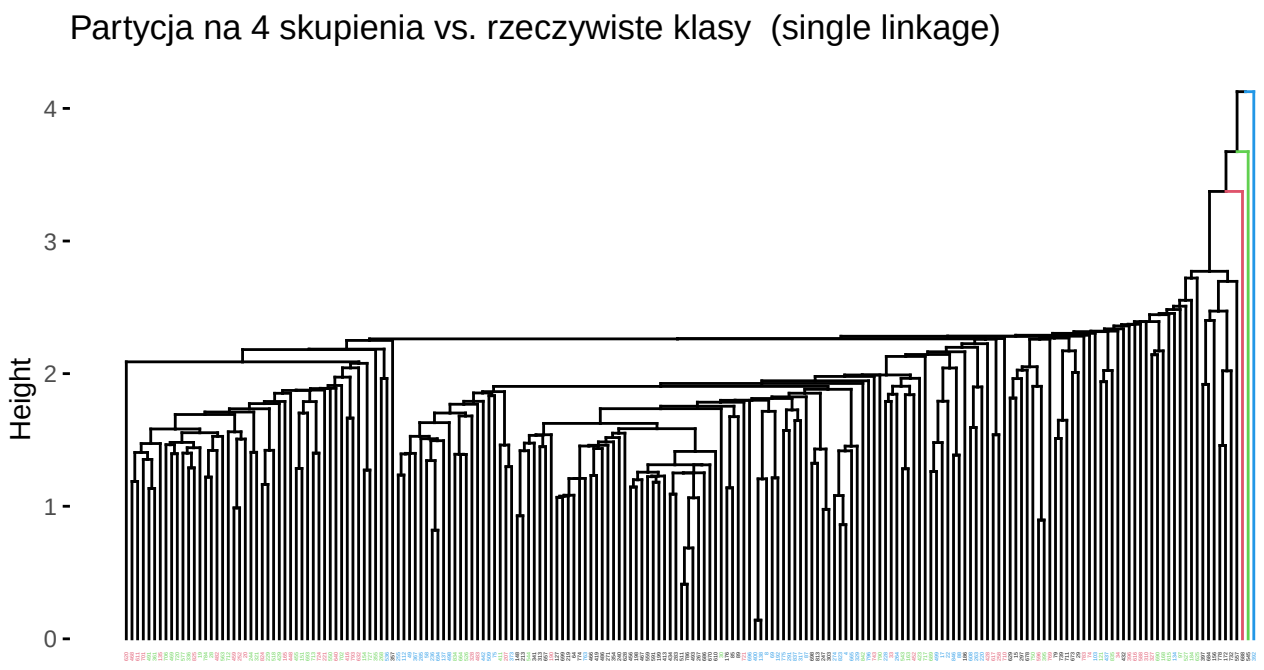
Wykres 19 przedstawia dendrogram uzyskany metodą łączenia `average linkage` z podziałem na 4 klastry. Widoczna jest dominacja klastra oznaczonego *zielonym* kolorem (zawiera on najwięcej obserwacji) i bardzo

mały wkład klastra *niebieskiego* (zawiera jedynie 2 obserwacje). Odcinając drzewo na poziomie czterech klastrów (zgodnie z liczbą klas), widzimy częściowe pokrycie z rzeczywistą strukturą danych. Nie jest ono jednak na wysokim poziomie. Przykładowo, dla klasy oznaczonej kolorem *czerwonym* żadna z obserwacji należących do tego klastra nie jest prawidłowo przypisana (obserwacje te powinny znajdować się w *czarnym* klastrze). Ta metoda dobrze radzi sobie z równomiernie rozproszonymi danymi i może wykrywać niekuliste skupienia. Jednakże dla danych Vehicle, które mają pewne klasy trudne do separacji (np. **"saab"** i **"opel"**), podział nie jest najlepszy.



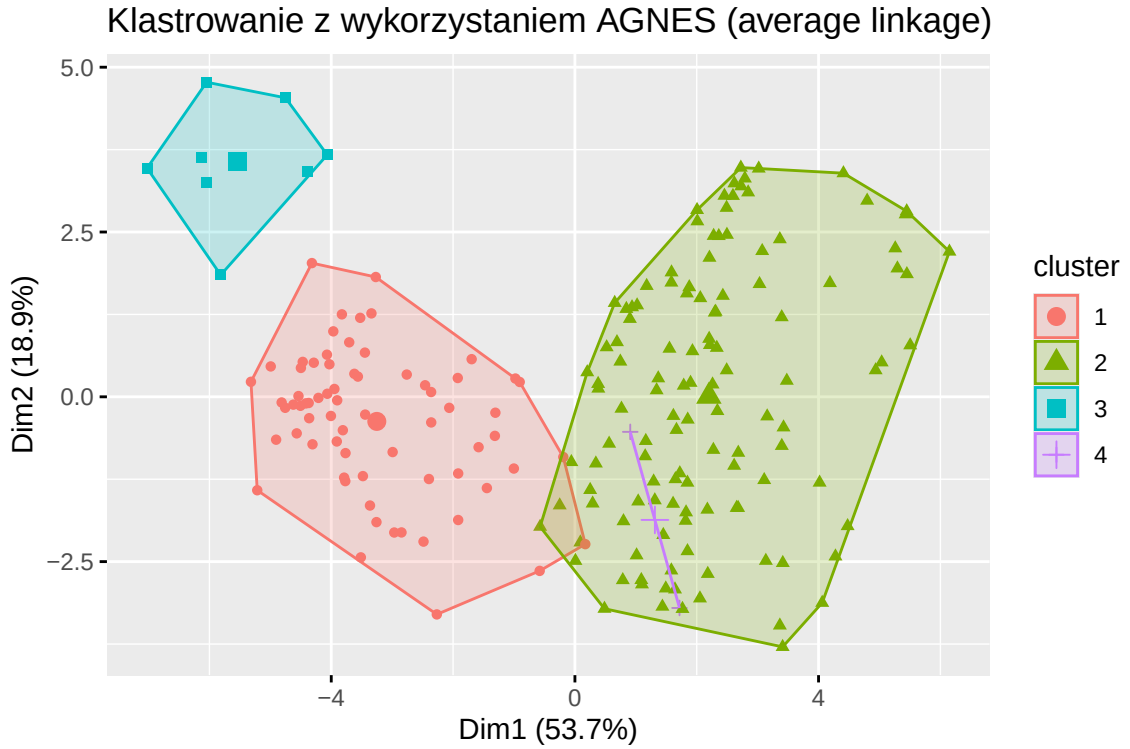
Wykres 20: Dendrogram z użyciem complete linkage

Dendrogram (wykres 20) uzyskany metodą `complete linkage` ma zdecydowanie bardziej równoliczne klastry. Można zauważyć, że metoda ta stworzyła dokładnie taki sam klaster (*czerwony*) jak w przypadku metody `average linkage` (zawiera te same obserwacje). Ponownie widoczna jest mała zgodność podziałów z rzeczywistymi etykietkami klas.



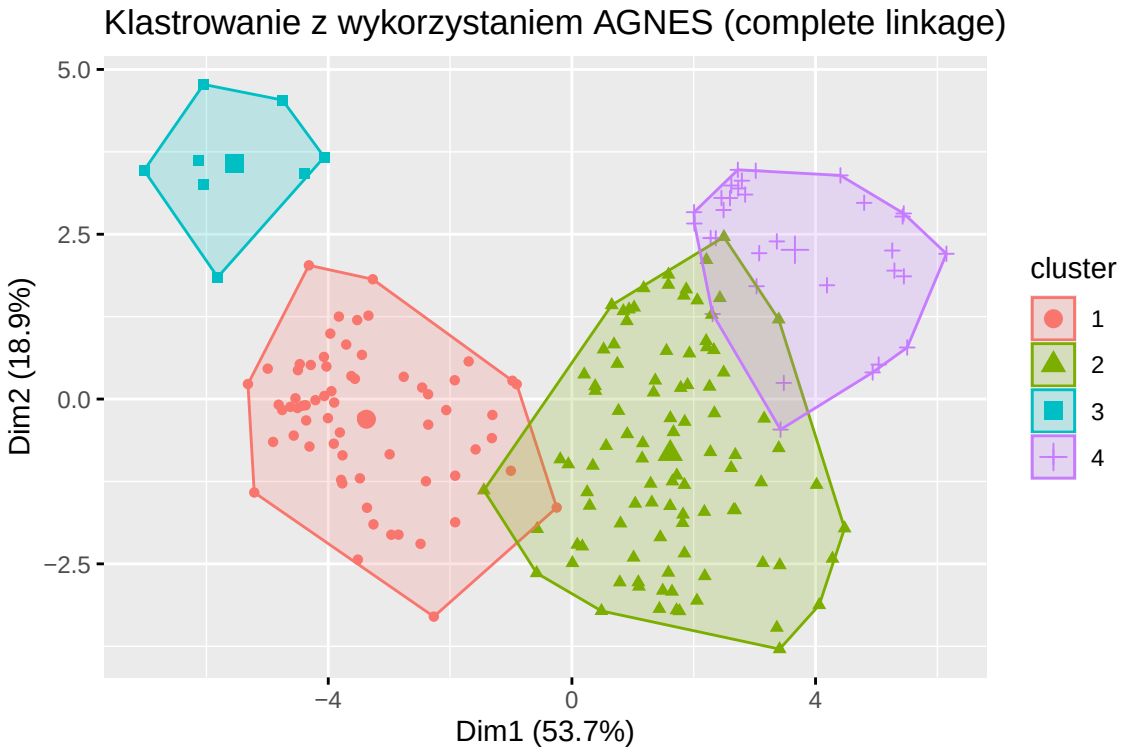
Wykres 21: Dendrogram z użyciem single linkage

Stosując metodę `single linkage` (wykres 21) otrzymujemy podziały tworzące efekt "łańcucha". Widzimy, że dane są łączone w sposób ciągły, często poprzez pojedyncze punkty, co prowadzi do bardzo wydłużonych i nierównych gałęzi. Taki układ nie sprzyja tworzeniu sensownych, zwartych skupień (wiele klas jest "zlepionych" w jeden duży klaster).



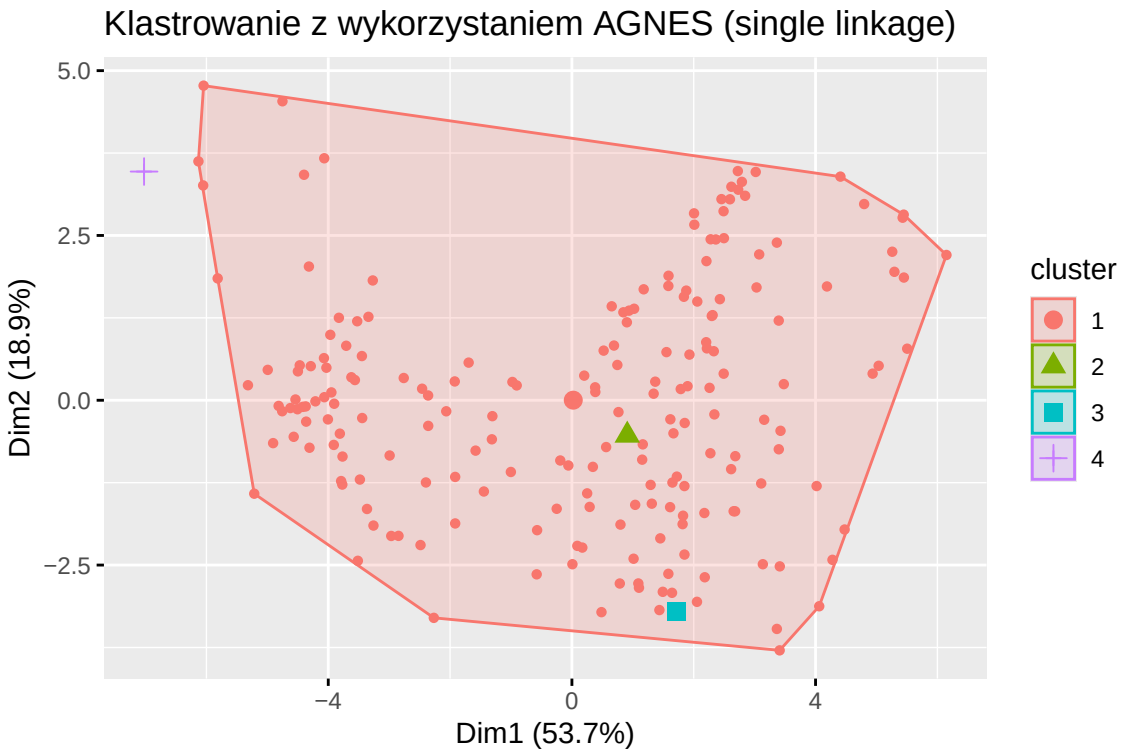
Wykres 22: Rzutowanie klastrów uzyskanych metodą AGNES (average linkage) na wykres PCA

Po odwzorowaniu wyników grupowania `AGNES` metodą `average linkage` na przestrzeń dwóch głównych składowych PCA widzimy, że niektóre klastry są wyraźnie wyodrębnione i tworzą zwarte grupy (np. klaster **3**). Inne skupiska są mniej jednoznaczne – punkty przypisane do różnych klastrów częściowo się przenikają (rzut klastra **4** znajduje się w całości wewnątrz rzutu klastra **2**), szczególnie w środkowej części wykresu. Wskazuje to, że nie wszystkie rzeczywiste klasy zostały odtworzone w sposób poprawny przez `AGNES`.



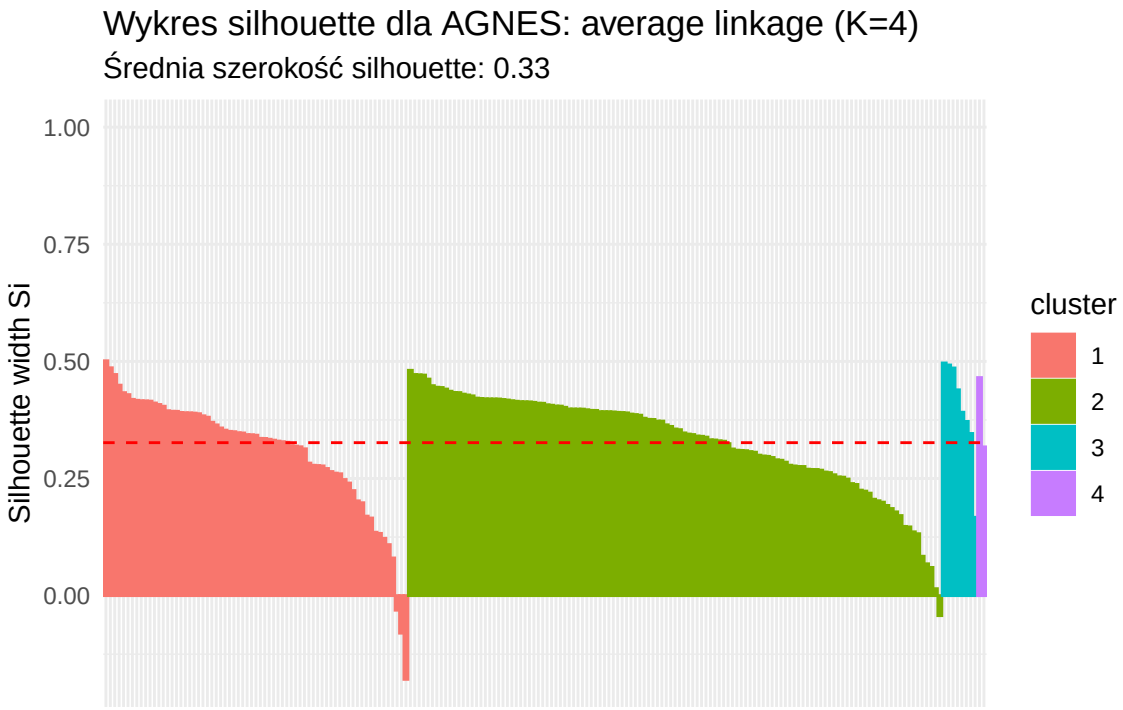
Wykres 23: Rzutowanie klastrów uzyskanych metodą AGNES (complete linkage) na wykres PCA

Wyniki dla `complete linkage` (wykres 23) są podobne do `average linkage`, ale klastry wydają się nieco bardziej zwarte. Mimo to nadal występuje silne nakładanie się rzutów klastrów (ponownie, jak przy `average linkage` klastra **1** i **2**). Wykres pokazuje, że nawet `complete linkage` nie rozwiązuje problemu słabej separacji klas w tym zbiorze danych.



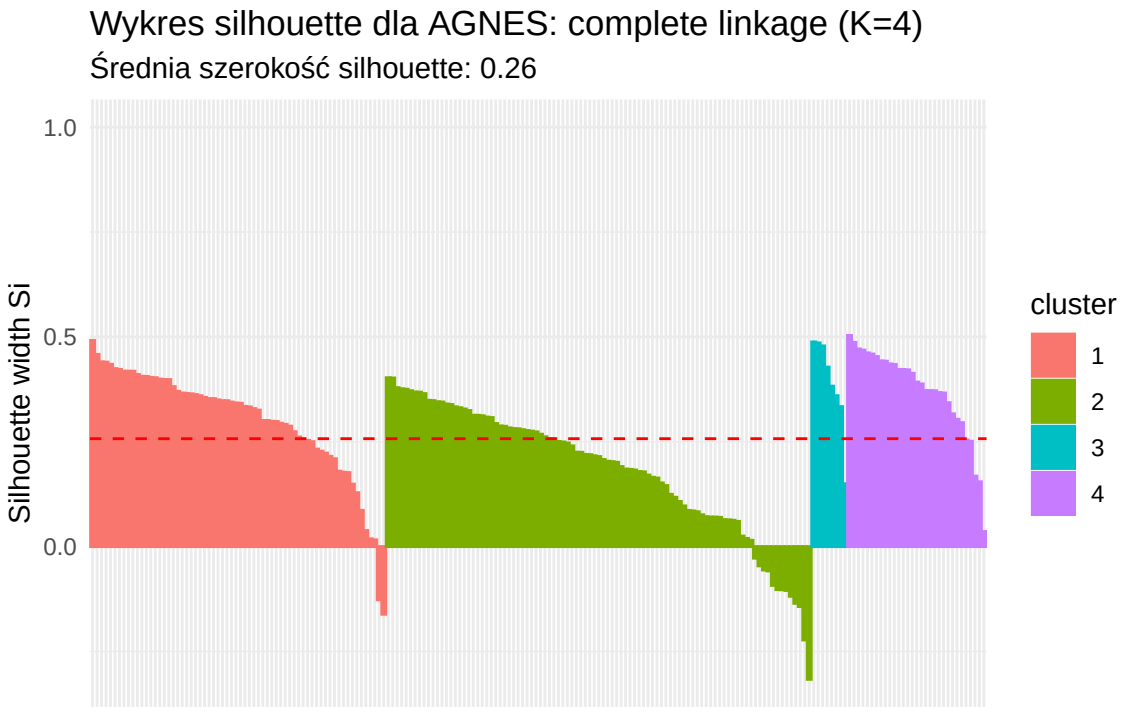
Wykres 24: Rzutowanie klastrów uzyskanych metodą AGNES (single linkage) na wykres PCA

Wykres dla `single linkage` (24) pokazuje jeszcze gorszą separację klastrów niż w poprzednich metodach. Niemal wszystkie obserwację są zgrupowane w jednym dużym klastrze, co potwierdza, że `single linkage` nie jest odpowiednią metodą dla tego typu danych. Wykres ilustruje, że metoda ta nie jest w stanie wykryć subtelnych różnic między klasami.



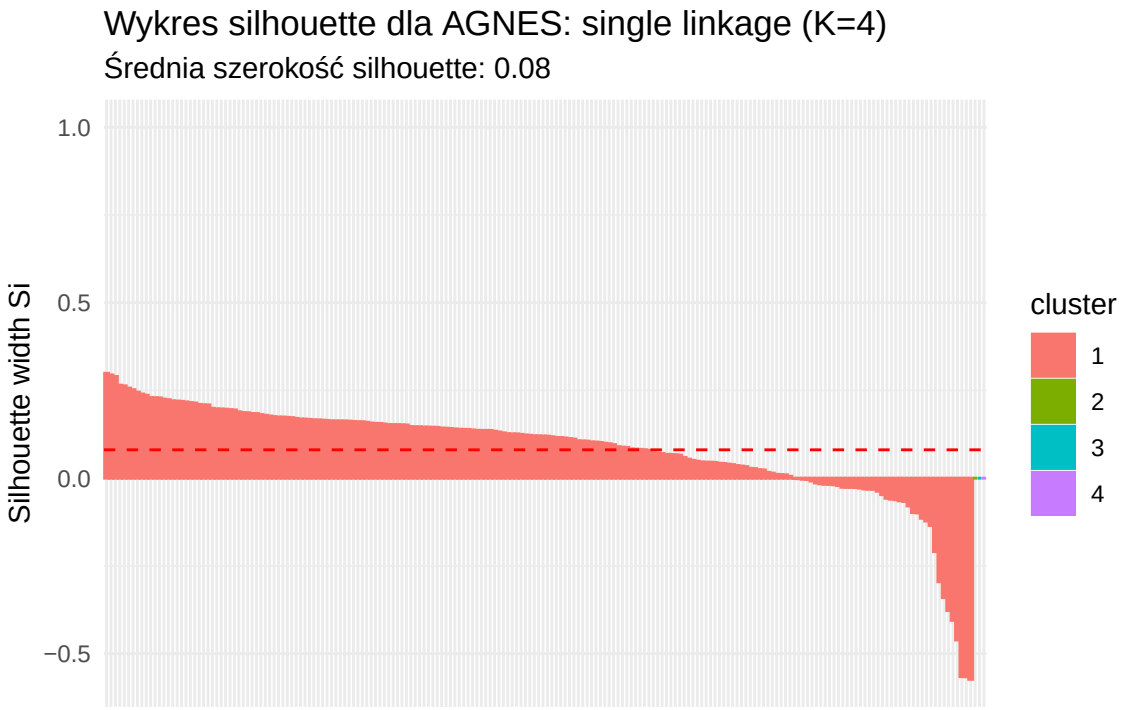
Wykres 25: Silhouette AGNES (average linkage), K=4, avg.width = 0.33

Średnia wartość silhouette dla metody `average linkage` (wykres 25) wynosi **0.33**, co jest wynikiem wyższym niż dla *k-means* i *PAM*. Wszystkie obserwajce z klastra **3** mają wartości powyżej średniej, co wskazuje na dobrą spójność, podczas gdy dla innych klastrów ledwie połowa obserwacji ma wartości powyżej średniej. Pozostałe mają niższe wartości (a nawet ujemne). Wykres sugeruje, że `average linkage` może być lepszy od metod grupujących, ale nadal nie jest idealny.



Wykres 26: Silhouette AGNES (complete linkage), K=4, avg.width = 0.26

Dla metody `complete linkage` średnia wartość silhouette wynosi **0.26** (wykres 26), co jest wartością niższą niż dla `average linkage`. Klastry **3** i **4** nadal mają wysoką wartość, ale pozostałe klastry są słabo zdefiniowane. Najgorsze wyniki otrzymujemy dla klastra **2**, gdzie znacząca część obserwacji została źle podporządkowana (ujemna wartość silhouette).

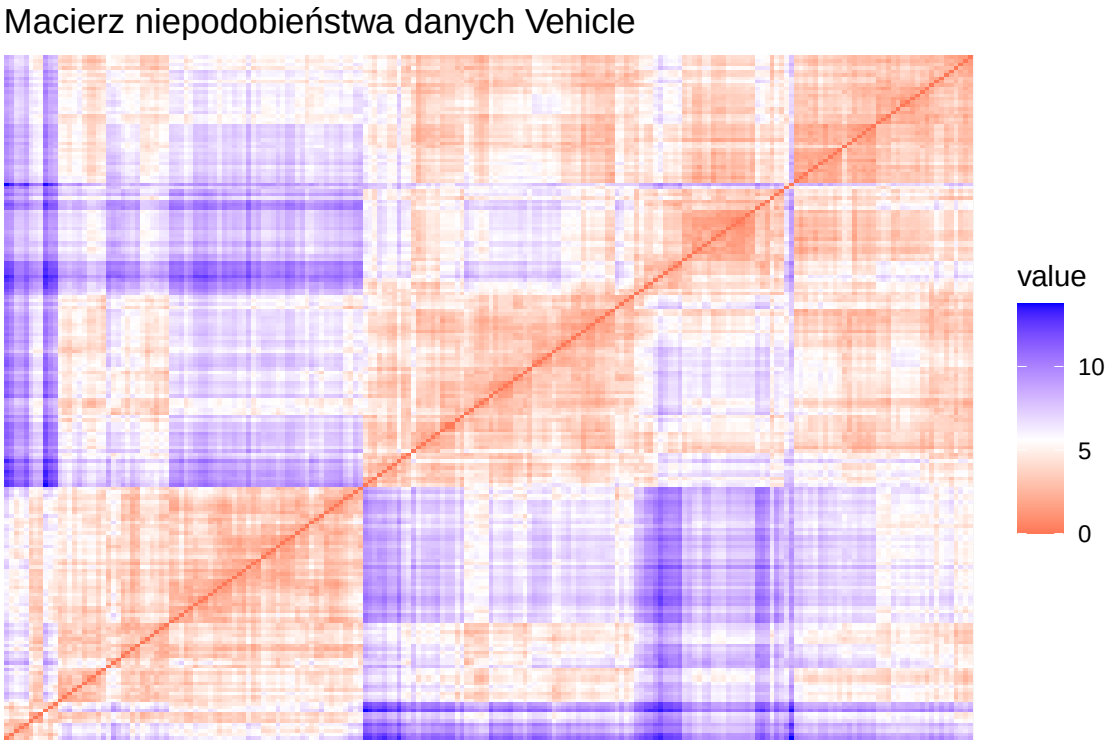


Wykres 27: Silhouette AGNES (single linkage), K=4, avg. width = 0.08

Dla metody łączenia `single linkage` średnia wartość silhouette wynosi zaledwie **0.08**, co jest najgorszym spośród wszystkich wyników. Większość obserwacji ma wartości bliskie zeru lub ujemne, co wskazuje na bardzo słabe grupowanie. Wykres 27 potwierdza, że single linkage jest najmniej skuteczną metodą dla tych danych.

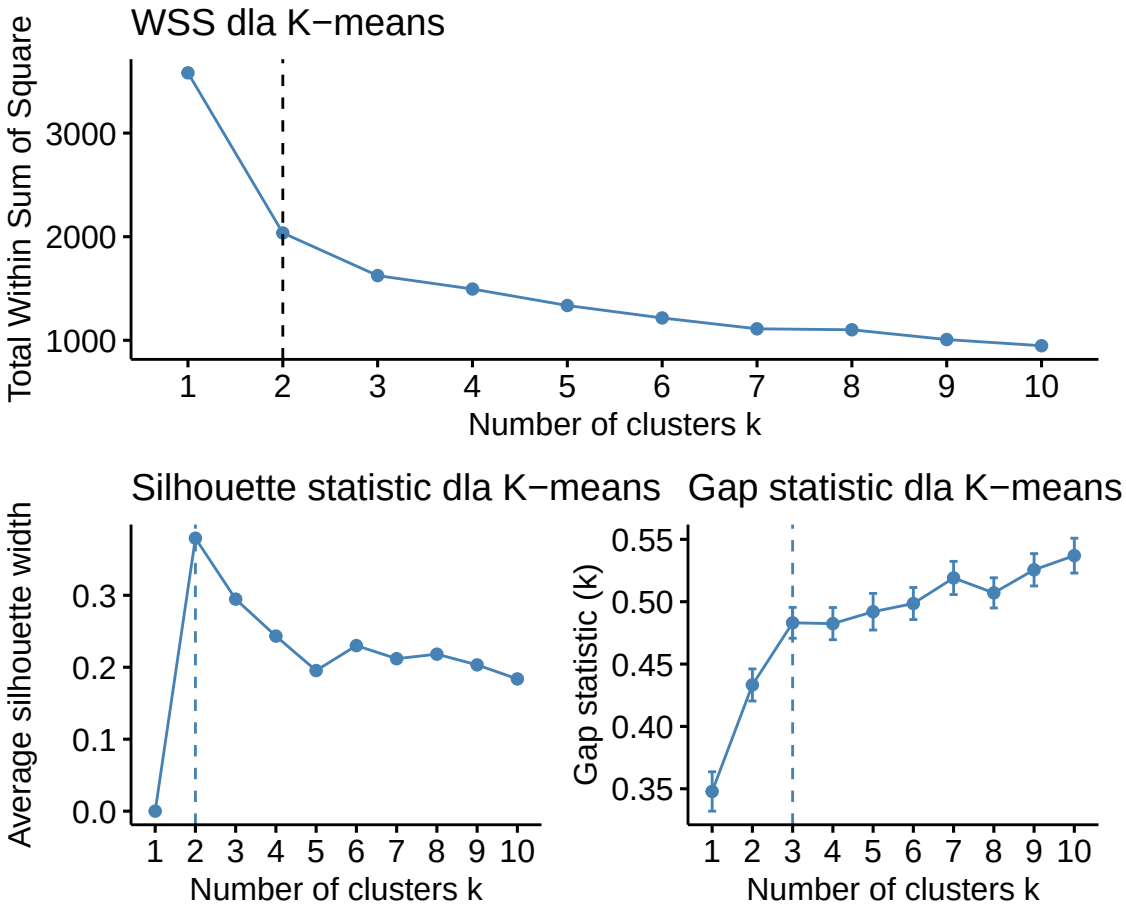
2.3 Ocena jakości grupowania. Wybór optymalnej liczby skupień i porównanie metod

W celu oceny jakości uzyskanych grupowań porównamy przypisania obserwacji do klastrów z rzeczywistymi etykietkami klas w zbiorze `Vehicle_scaled`. Dodatkowo wykorzystamy różne wskaźniki zewnętrzne oraz wewnętrzne w celu wyboru optymalnej liczby skupień.



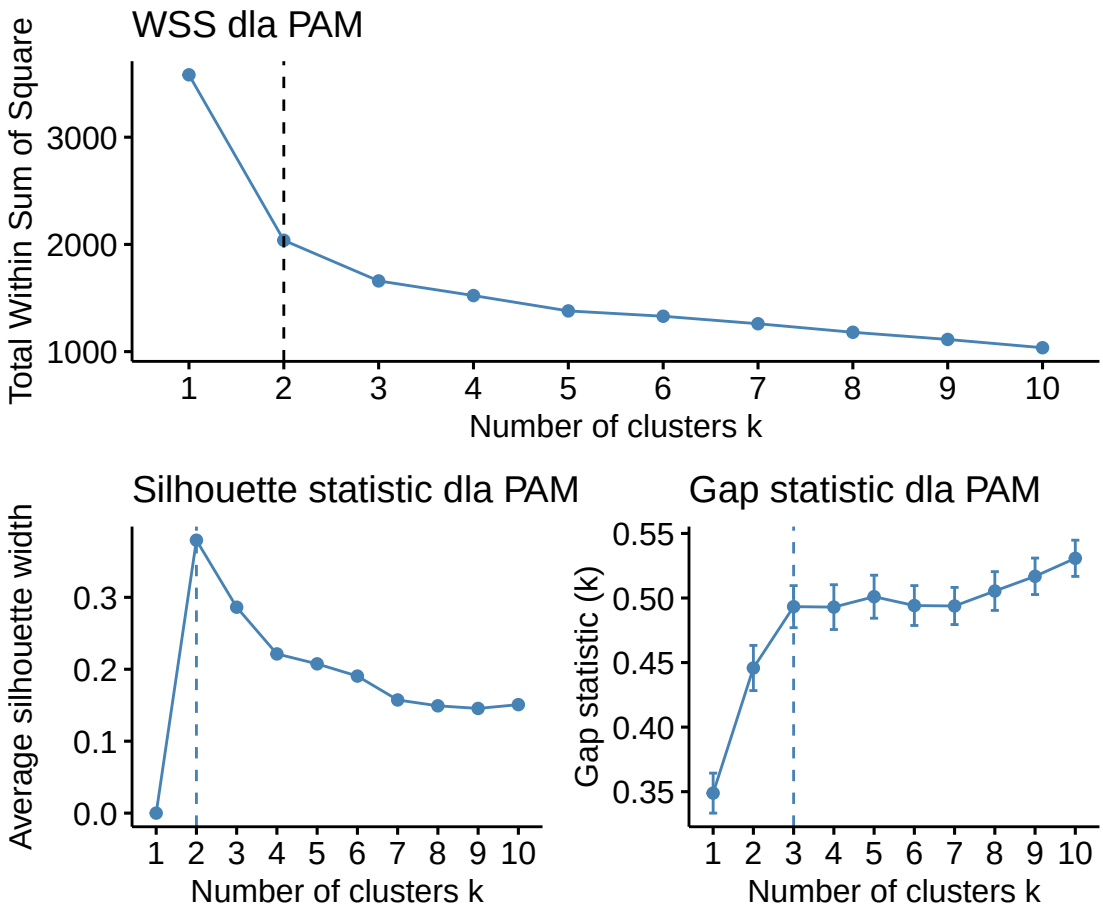
Wykres 28: Macierz niepodobieństwa danych Vehicle (uporządkowana)

Macierz niepodobieństwa (wykres 28) pokazuje odległości między obserwacjami. Czerwone obszary oznaczają małe odległości, a niebieskie – duże. Widoczne są pewne grupy o podobieństwach. Możemy subiektywnie wyznaczyć cztery główne podziały (czerwone kwadraty), które można zauważyć wzdłuż głównej przekątnej. Brakuje jednak wyraźnych granic między klastrami. Wykres potwierdza, że dane są trudne do separacji, co tłumaczy niską skuteczność metod grupowania.



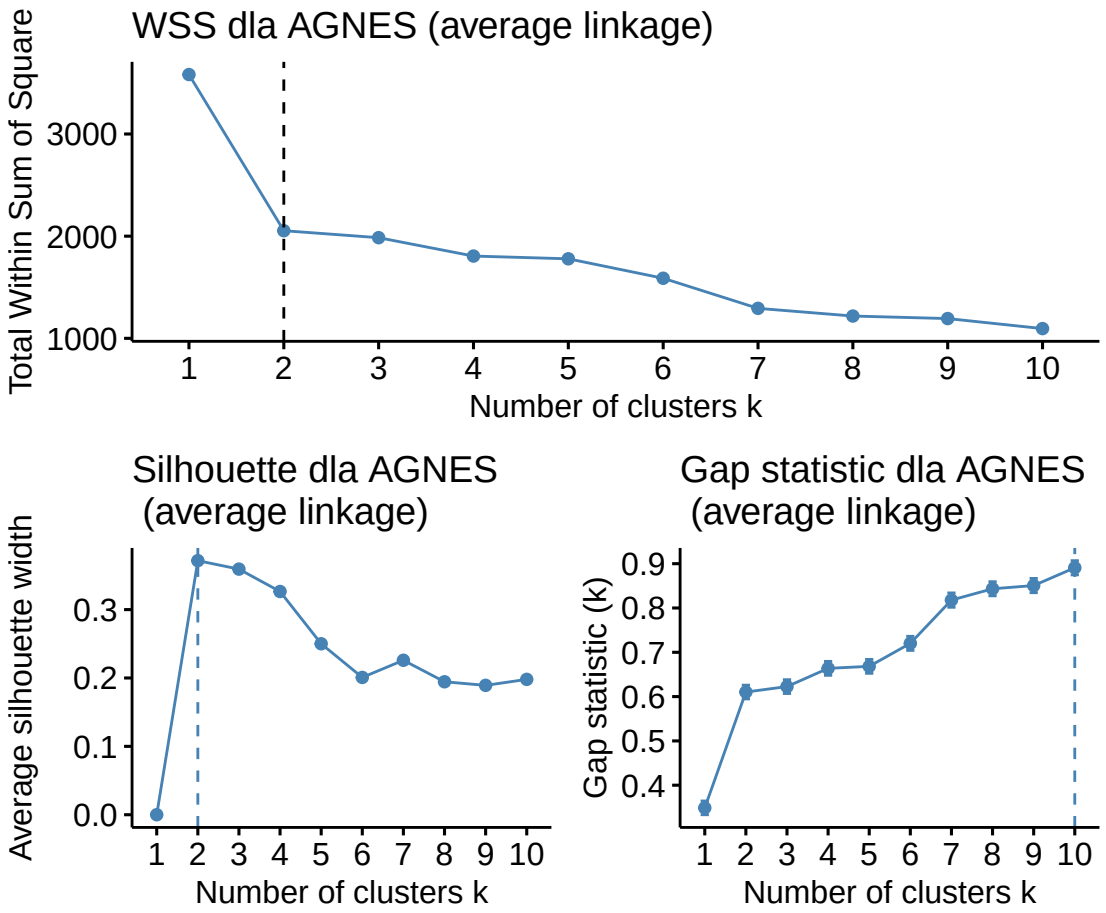
Wykres 29: Wybór optymalnej liczby klastrów: wss, silhouette i gap statistic (k-means)

Wykres (29) `WSS` (Within Sum of Squares) pokazuje, że wartość błędu maleje wraz ze wzrostem liczby klastrów, ale spadek jest znacznie wolniejszy po **K=2**, co sugeruje, że **2** klastry mogą być rozsądnym wyborem. Wykres `Silhouette` wskazuje na optymalną liczbę **K=2** (najwyższa średnia wartość). `Gap Statistic` potwierdza **K=3** jako najbardziej optymalny podział. Wyniki te sugerują, że wybór **K=2** jest kompromisem między redukcją WSS a zachowaniem spójności klastrów.



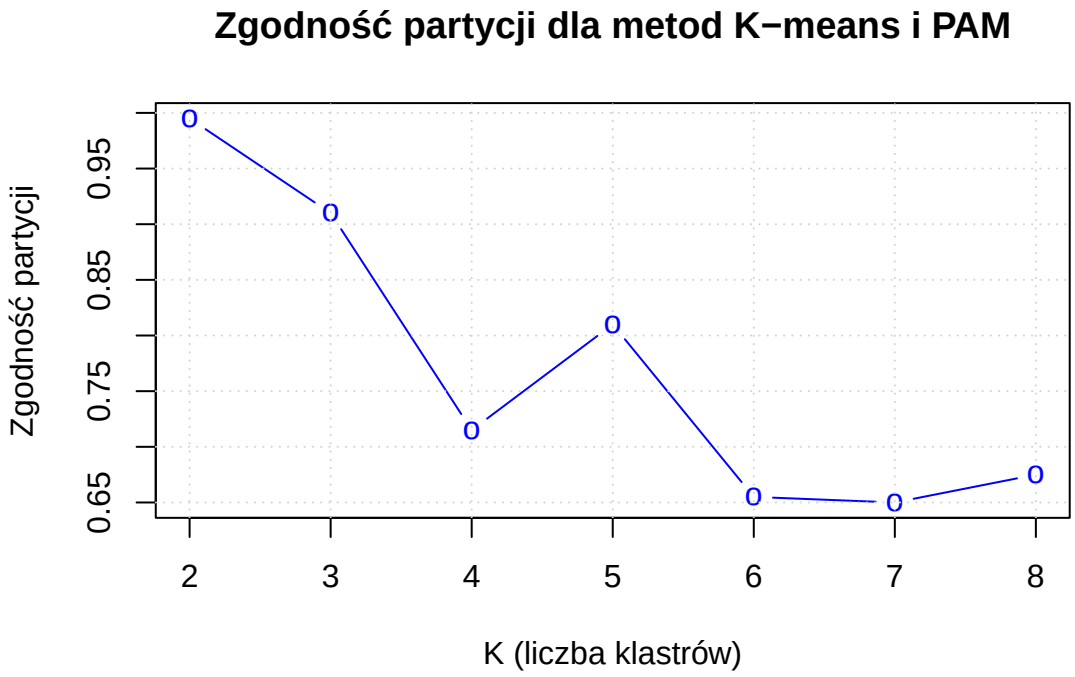
Wykres 30: Wybór optymalnej liczby klastrów: wss, silhouette i gap statistic (PAM)

Wykres 30 dla `PAM` prezentują niemal identyczne wyniki co te dla `K-means`. Ponownie **K=2** wydaje się być najlepszym wyborem.



Wykres 31: Wybór optymalnej liczby klastrów: wss, silhouette i gap statistic (AGNES)

Dla algorytmu `AGNES` z metodą łączenia `average linkage` dla WSS punkt łokciowy występuje w **K=2**, podobnie jak wcześniejsze metody. `Silhouette` osiąga szczyt dla **K=2**. `Gap Statistic` wskazuje na **K=10**. Metoda `AGNES` wydaje się bardziej stabilna niż `PAM`, ale mniej precyzyjna niż `K-means` w identyfikacji naturalnych klastrów.

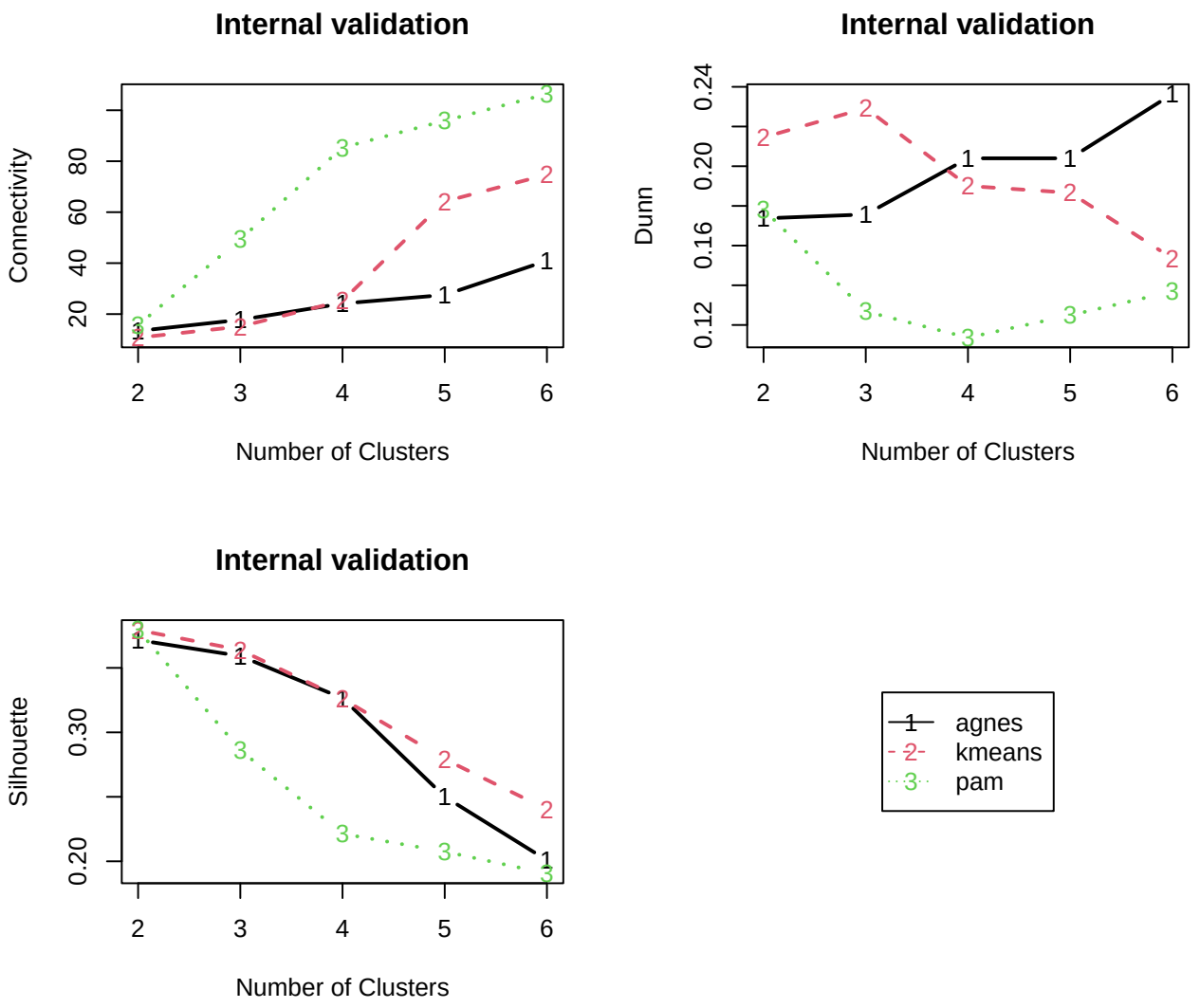


Wykres 32: Zgodność partycji uzyskanych metodami k-means i PAM w zależności od liczby klastrów K

2.3.1 Wskaźniki wewnętrzne

Skorzystamy z funkcji *clValid* w celu wyznaczenia wartości wskaźników wewnętrznych w zależności od liczby klastrów. Przeanalizujemy trzy indeksy:

- **Silhouette** - bada zwartość klastrów i separację przestrzenną,
- **Wskaźnik Dunn'a**,
- **Connectivity** - bada spójność i zdolność przyłączeniową.

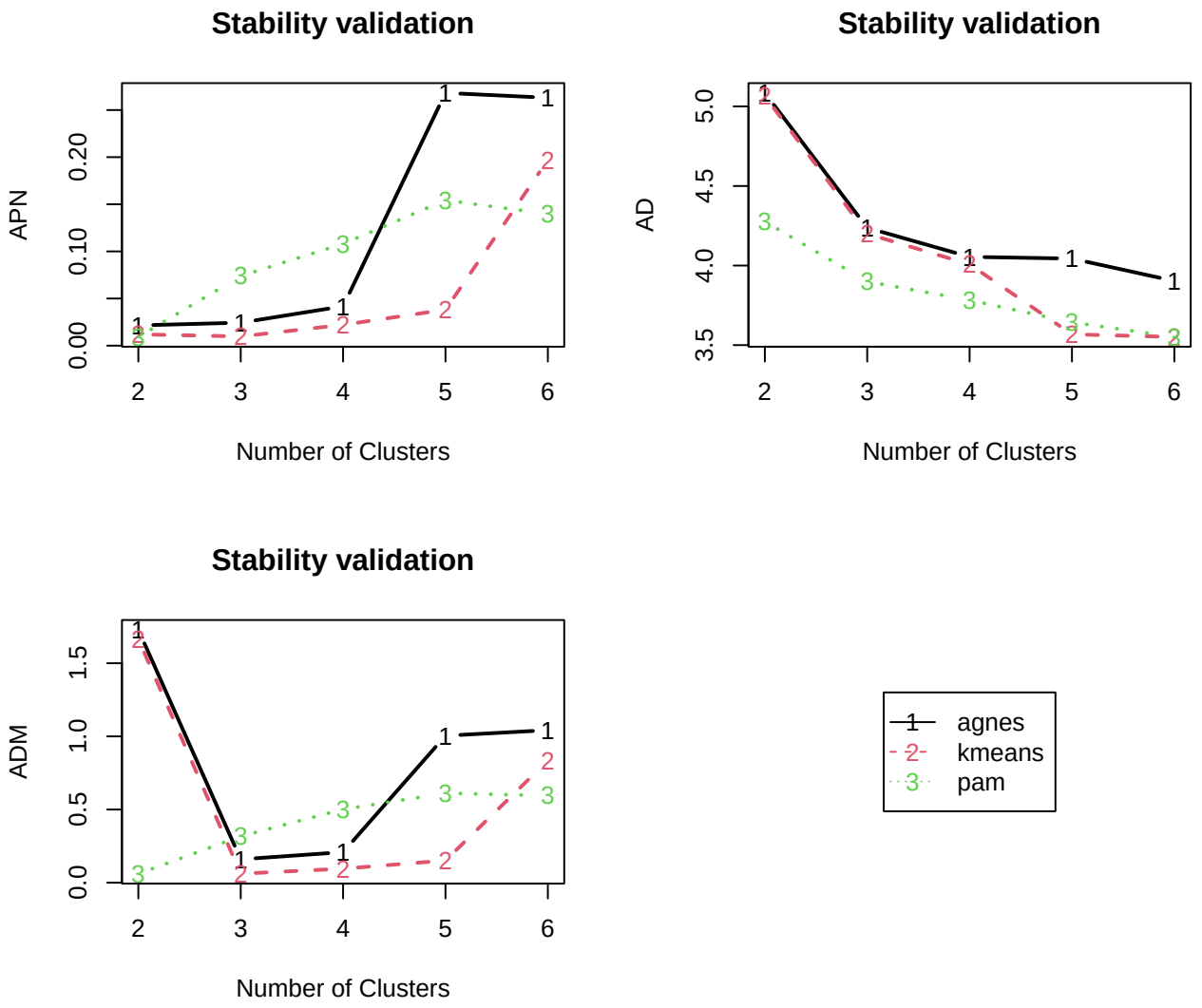


Wykres 33: Porównanie metod klasteryzacji wg wewnętrznych indeksów jakości (2–6 klastrów)

2.3.2 Ocena stabilności

Ponownie wykorzystujemy funkcję *clValid* aby zbadać stabilność algorytmów w zależności od liczby podziałów. Przeanalizujemy trzy miary:

- **APN** – Average Proportion of Non-overlap,
- **AD** – Average Distance,
- **ADM** – Average Distance between Means.



Wykres 34: Ocena stabilności metod klasteryzacji na podstawie miar APN, AD i ADM

Wskaźniki stabilności (APN, AD, ADM) pokazują, że `k-means` jest najbardziej stabilny dla **K=3** (zwłaszcza dla wskaźnika ADM i APN wartość ta jest najmniejsza spośród wszystkich). `PAM` ma niższą stabilność, szczególnie dla **K>2**, lecz skoki wartości są między kolejnymi podziałamu są zdecydowanie łagodniejsze. `AGNES` zachowuje stabilność podobną do `k-means`, ale dla **K>4** wyniki są gorsze. Podsumowując wykresy 34, `k-means` wydaje się być najbardziej stabliną metodą spośród omawianych.

2.3.3 Wybór najlepszej liczby klastrów na podstawie wielu wskaźników

Wykorzystujemy funkcję `NbClust` z pakietu *NbClust*, która zawiera wiele wskaźników

```
set.seed(1025)
res <- NbClust(Vehicle_scaled, diss = NULL, distance = "euclidean",
    min.nc = 2, max.nc = 6, method = "kmeans", index = "all")
```

```
etykietki_NbClust <- res$Best.partition

tab_nbclust <- table(etykietki_NbClust, Vehicle_etykiety)
```

Funkcję *NbClust* można wykorzystać również do wyznaczenia podziału zaproponowanego przez tę funkcję.

```
etykietki_NbClust <- res$Best.partition
tab_nbclust <- table(etykietki_NbClust, Vehicle_etykiety)

matchClasses(tab_nbclust)
```

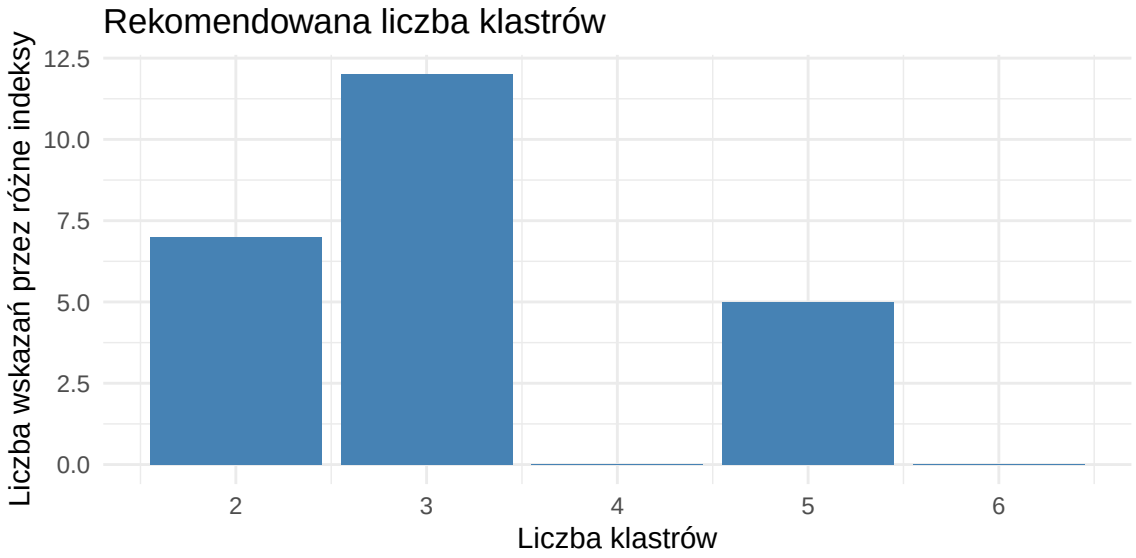
```
## Cases in matched pairs: 42.5 %
```

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

```
(dokl_nbclust <- compareMatchedClasses(etykietki_NbClust,
    Vehicle_etykiety)$diag)
```

```
##           [,1]
## [1,] 0.5345912
```

Używając funkcji *compareMatchedClasses* uzyskujemy zgodność na poziomie 53.5%.



Wykres 35: Rekomendowana liczba klastrów według różnych indeksów jakości

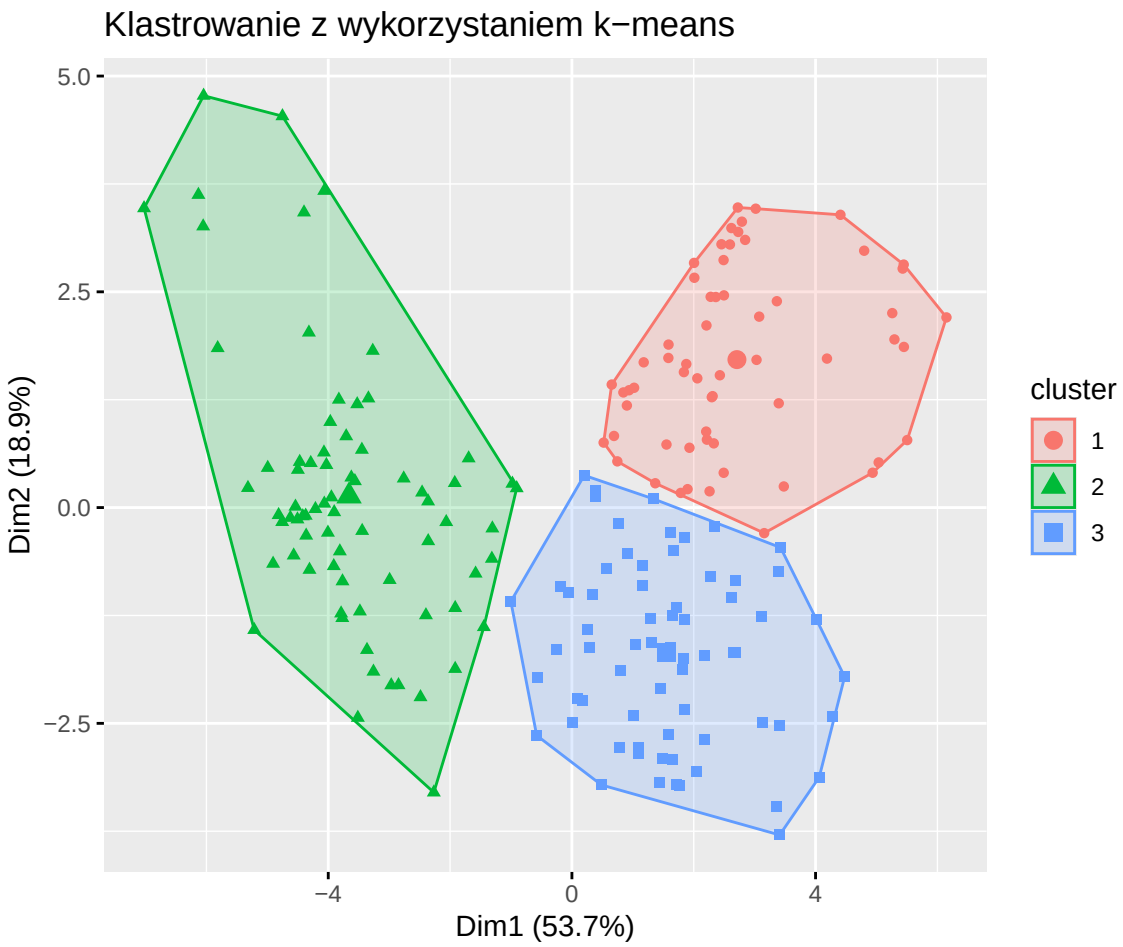
Większość indeksów (wykres 35) rekomenduje **K=2** lub **K=3** (dla 3 klastrów aż 12 sposród 24 wskaźników wskazało taki podział). Brak zgodności między indeksami potwierdza, że zbiór danych jest trudny do jednoznacznego podziału.

Podsumowując wszystkie wnioski, wybieramy **K=3** jako optymalny podział, gdyż został on wskazany przez największą liczbę wskaźników.

2.4 Interpretacja wyników grupowania

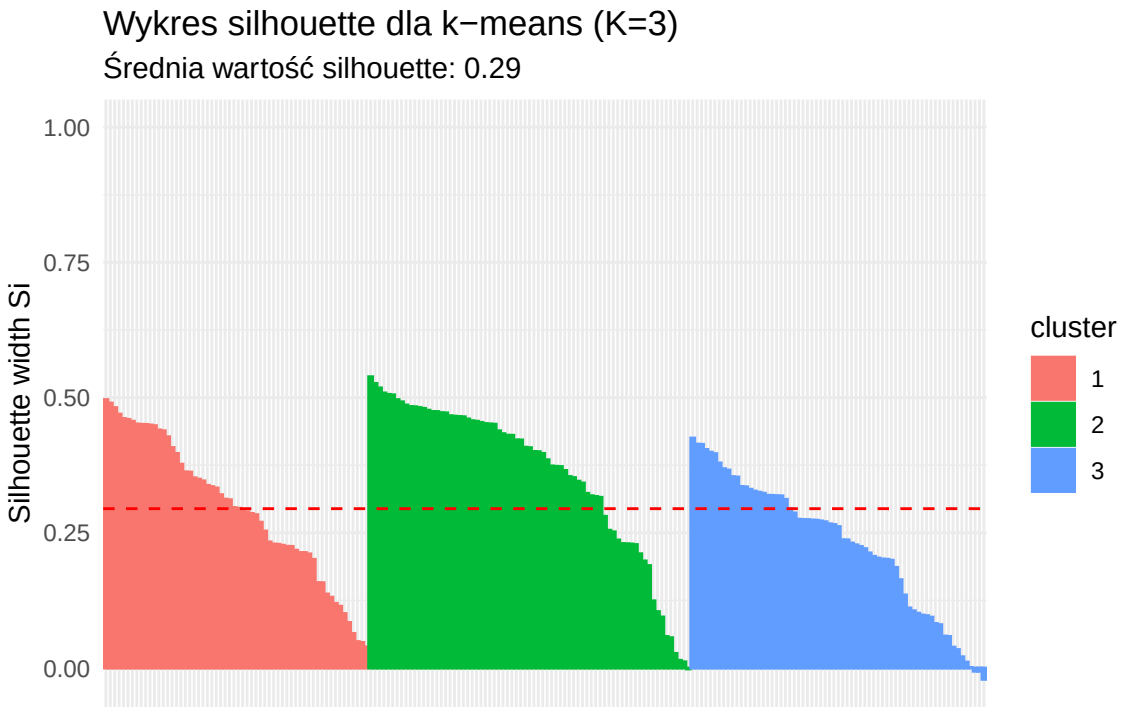
Przeprowadzamy analogiczną analizę dla **K=3**, wykorzystując metody `k-means`, `PAM` oraz `AGNES`.

2.4.1 K-means dla K=3



Wykres 36: Klastry uzyskane metodą k-means w PCA

Na wykresie 36 można zauważyć wyniki grupowania `k-means` w przestrzeni dwóch pierwszych składowych głównych. Możemy zauważyć dobrą separację wszystkich rzutów klastrów. Dodatkowo, klastry te są dość zwarte i wypukłe.



Wykres 37: Silhouette dla k-means

Wykres `silhouette` dla algorytmu *k-means* (37) pokazuje wyższą średnią wartość wskaźnika (**0.29**) niż dla czterech klastrów. Nieiczne obserwacje w klastrze **3** mają ujemne wartości `silhouette`, co oznacza, że zostały przypisane do niewłaściwego klastra lub znajdują się blisko granic między klastrami.

```
tab_vehicle_kmeans_3 <- table(Vehicle_scaled_etykietki_kmeans_optymalne_klastry,
    Vehicle_etykiety)

matchClasses(tab_vehicle_kmeans_3)
```

```
## Cases in matched pairs: 42.5 %
```

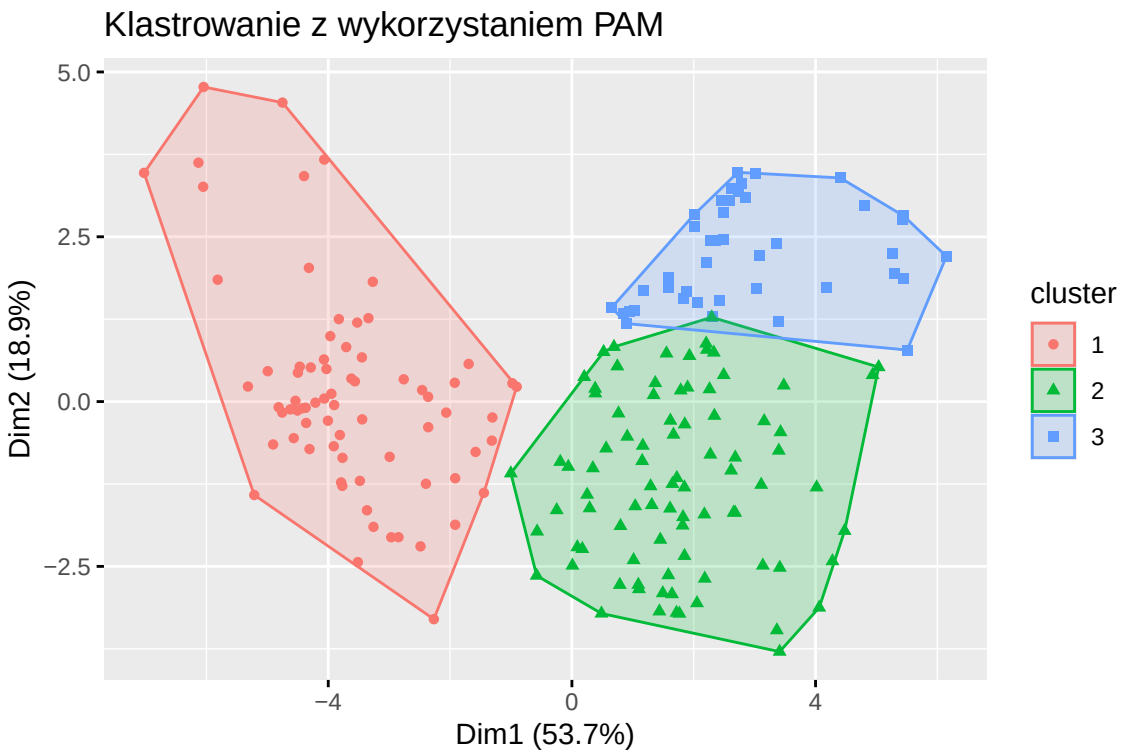
```
##      1      2      3
##  "bus" "saab"  "van"
```

```
(kmeans_dokl_3 <- compareMatchedClasses(Vehicle_scaled_etykietki_kmeans_optymalne_klastry,
    Vehicle_etykiety)$diag)
```

```
##           [,1]
## [1,] 0.5345912
```

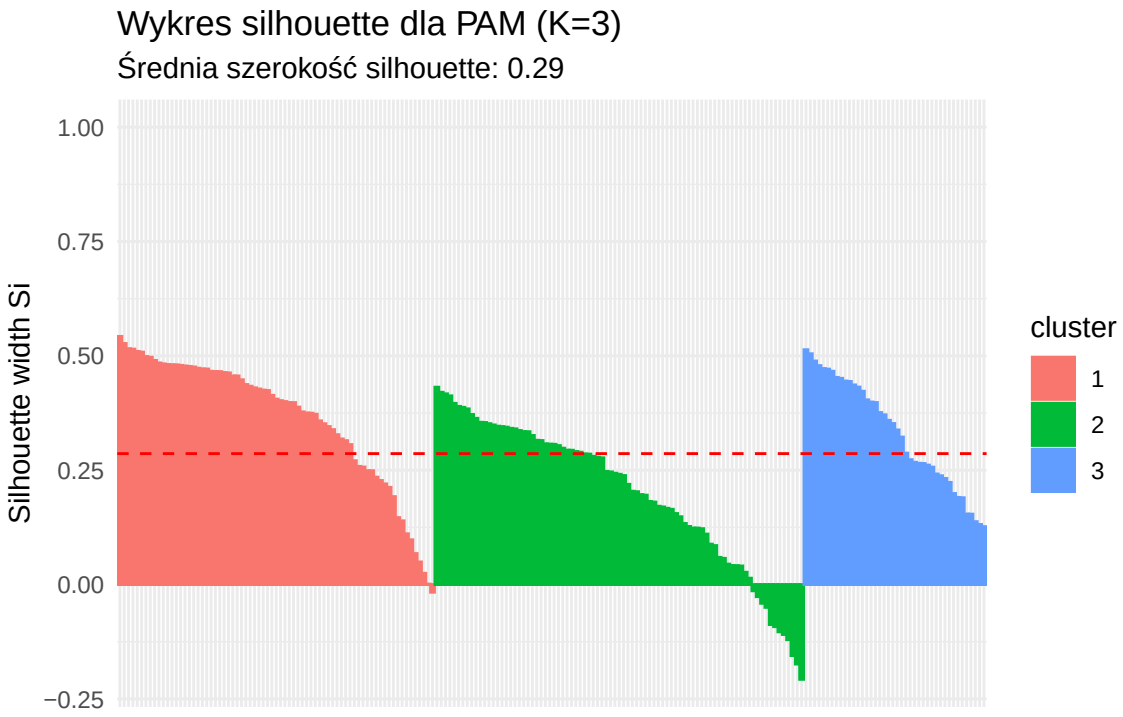
Korzystając z funkcji `compareMatchedClasses` uzyskujemy zgodność klasyfikacji na poziomie 53.5%.

2.4.2 PAM dla K=3



Wykres 38: Wizualizacja wyników PAM w przestrzeni PCA

Zrzutowanie wyników klasteryzacji metodą `PAM` (wykres 38) pokazuje dość gorszą separację rzutów klas niż w przypadku `k-means`. Klaster **1** pozostał bez zmian.



Wykres 39: Wykres wskaźnika silhouette dla metody PAM

Wykres `silhouette` dla algorytmu `PAM` 39 przedstawia identyczną średnią wartość tego wskaźnika (**0.29**) co dla *k-means*. Spora część obserwacji klastra **2** ma ujemną wartość wskaźnika `silhouette`. Obserwujemy jednak wyższe wartości dla klastrów **1** i **3**.

```
## Cases in matched pairs: 43.5 %
```

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

```
##           [,1]
## [1,] 0.5471698
```

Korzystając z funkcji `compareMatchedClasses` uzyskujemy zgodność klasyfikacji na poziomie 54.7%. Jest to wynik lepszy o nieco ponad 1 punkt procentowy niż dla `k-means`.

2.4.3 AGNES dla K=3

Wykorzystamy tylko metodę łączenia `average linkage` i `complete linkage`, ponieważ dawały one najlepsze wyniki dla poprzedniej analizy.

```
Agnes_avg_Veh_3 <- agnes(x = mac_niepodob_Veh, diss = TRUE,
    method = "average")
Agnes_complete_Veh_3 <- agnes(x = mac_niepodob_Veh,
    diss = TRUE, method = "complete")
```

```
etykiety_agnes_avg_3 <- cutree(Agnes_avg_Veh, k = 3)

etykiety_agnes_comp_3 <- cutree(Agnes_complete_Veh_3,
    k = 3)
```

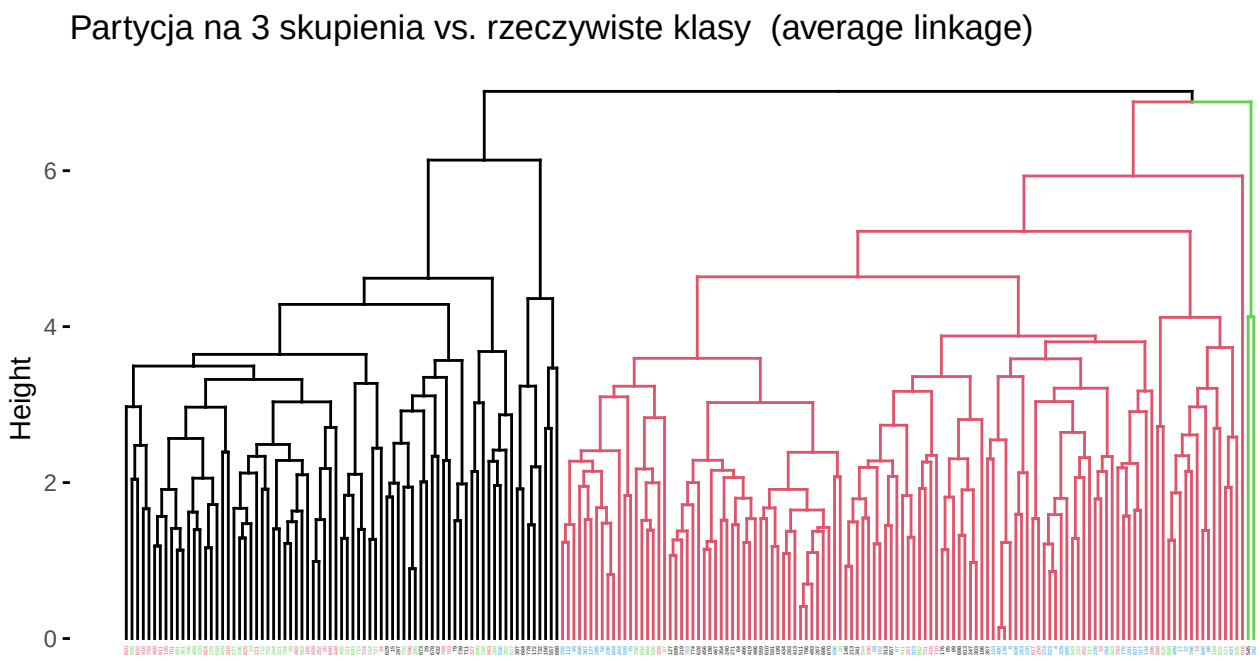
```r
# Sprawdzenie z rzeczywistymi klasami
tab_agnes_avg_3 <- table(etykiety_agnes_avg_3, Vehicle_etykiety)
(dokl_agnes_3_average <- (compareMatchedClasses(etykiety_agnes_avg_3,
    Vehicle_etykiety)$diag))
```

```
##           [,1]
## [1,] 0.490566
```

```r
tab_agnes_complete_3 <- table(etykiety_agnes_comp_3,
    Vehicle_etykiety)
(dokl_agnes_3_complete <- compareMatchedClasses(etykiety_agnes_comp_3,
    Vehicle_etykiety)$diag)
```

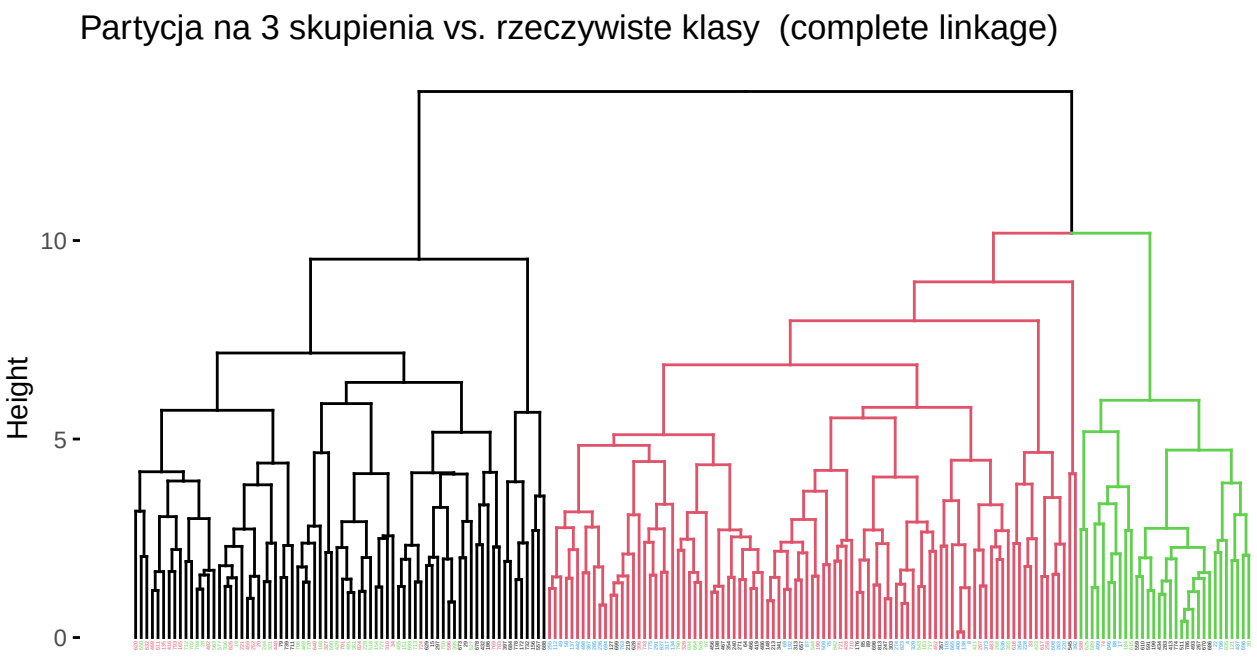
```
##           [,1]
## [1,] 0.5220126
```

Wykorzystując funkcję *compareMatchedClasses* otrzymujemy nieco lepszy wynik dla metody `complete linkage` (52.2%).



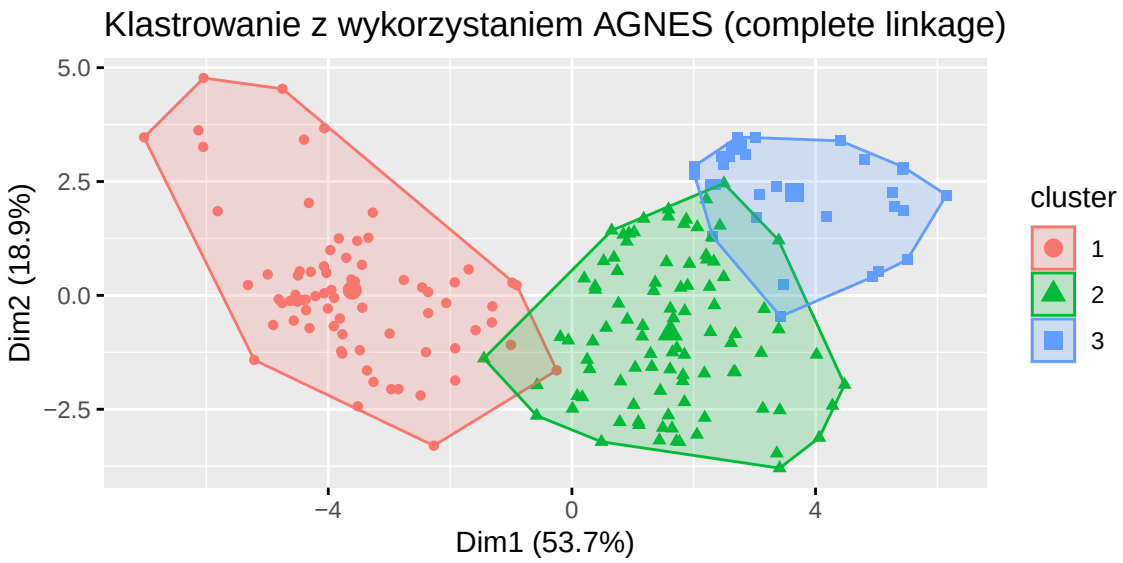
Wykres 40: Dendrogram z użyciem average linkage

Wykorzystując metodę (wykres 40) `average linkage` możemy zaobserwować dwa główne klastry (trzeci klaster od lewej zawiera jedynie dwie obserwacje).



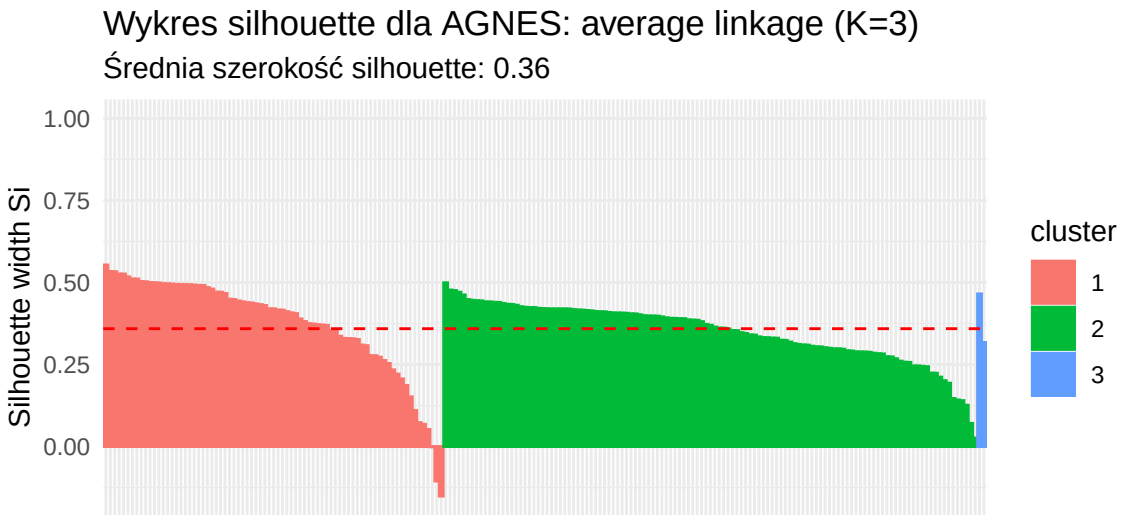
Wykres 41: Dendrogram z użyciem complete linkage

Korzystając z metody (wykres 40) `complete linkage` możemy zaobserwować, że do klastra *zielonego* zostało przydzielonych więcej obserwacji.



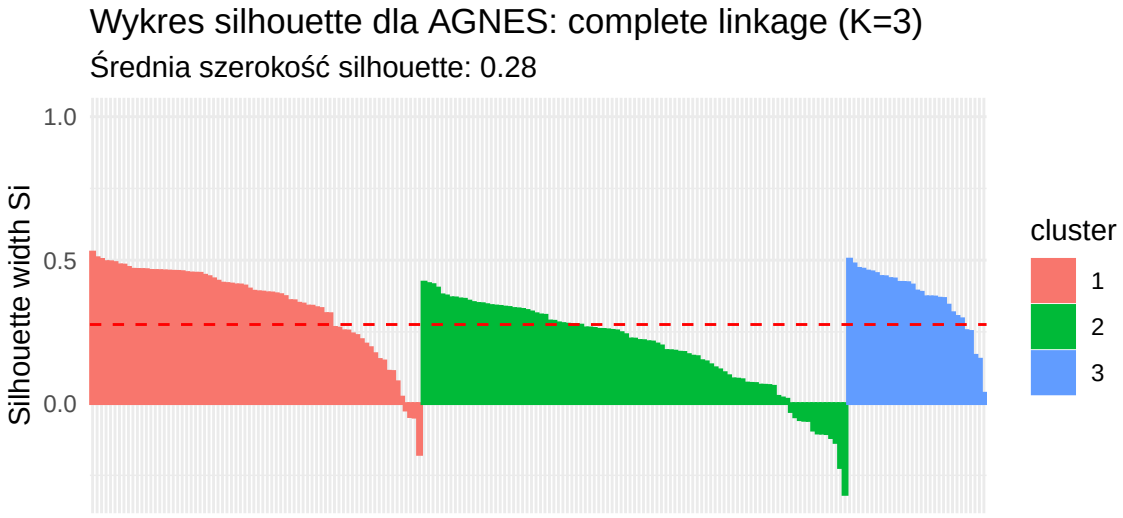
Wykres 42: Rzutowanie klastrów uzyskanych metodą AGNES (complete linkage) na wykres PCA

Wyniki dla `complete linkage` (wykres 42) możemy zauważyć częściowe nakładanie się rzutów klastrów. Separacja przestrzenna jest gorsza niż dla `k-means` i `PAM`.



Wykres 43: Silhouette AGNES (average linkage), K=3, avg.width = 0.36

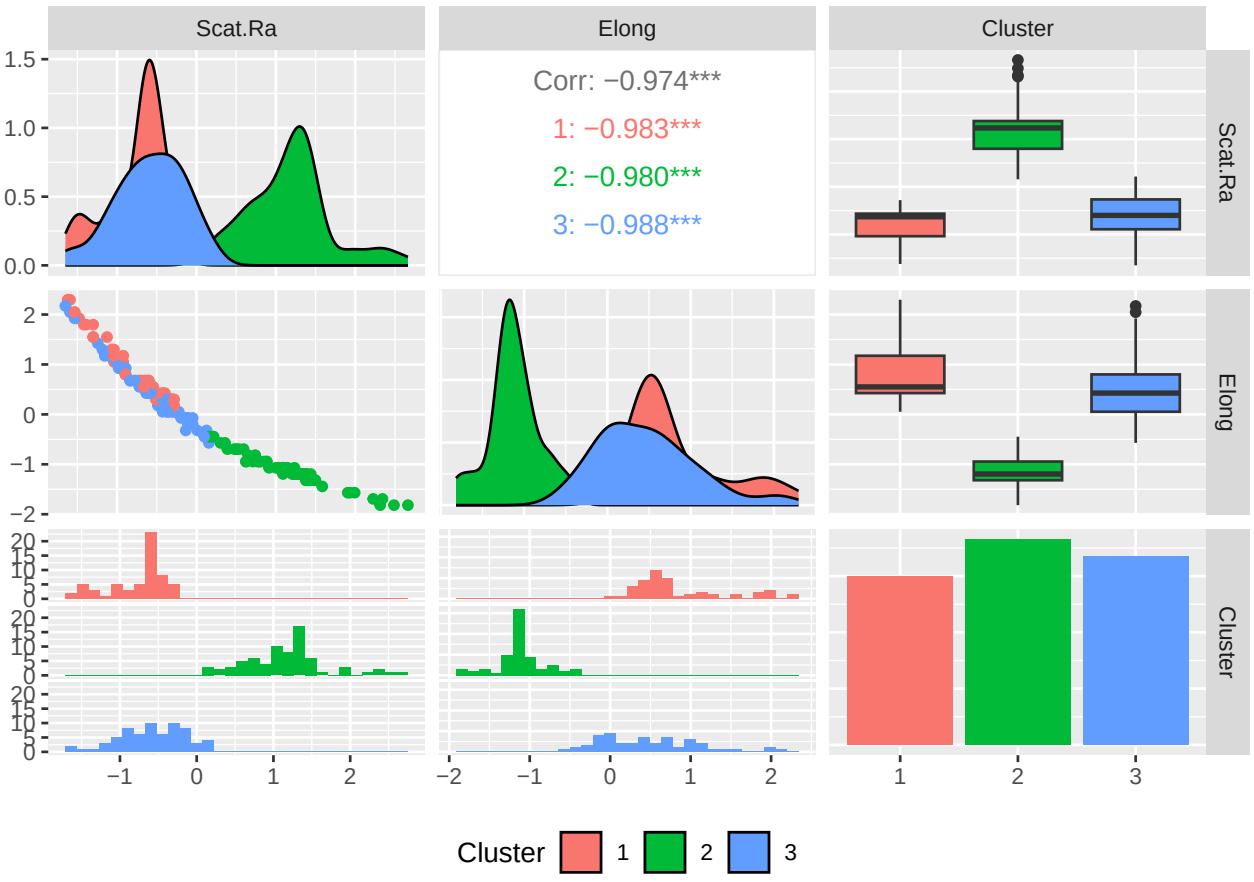
Średnia wartość silhouette dla metody `average linkage` (wykres 25) wynosi **0.36**, co jest najwyższym uzyskanym dotąd wynikiem. Niemal wszystkie obserwajce z klastra **3** mają wartości powyżej średniej, co wskazuje na dobrą spójność. Spora część obserwacji z klastra **2** ma wartości ujemne (w tym klastrze przyjmowane wartości `silhouette` są też niższe niż w pozostałych klastrach).



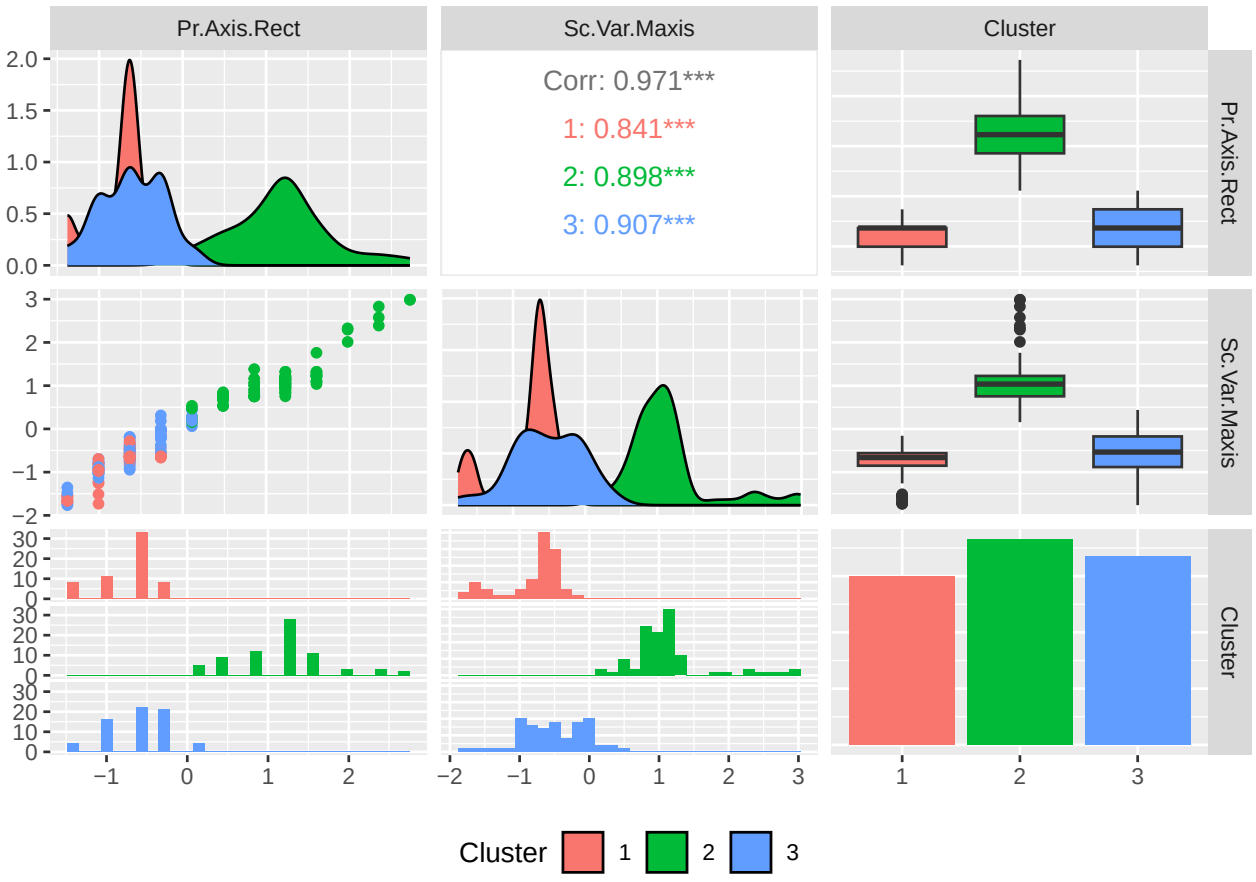
Wykres 44: Silhouette AGNES (complete linkage), K=3, avg.width = 0.28

2.4.4 Szukanie cech wyróżniających klastry

Wybieramy metodę `k-means` dla **K=3** i stosujemy funkcję *ggpairs* aby stworzyć zestawienie wybranych zmiennych w celu wyszczególnienia cech, które wyróżniają dane klastry.



Wykres 45: Zestawienie wybranych cech z podziałem na klastry (K=3)



Wykres 46: Zestawienie wybranych cech z podziałem na klastry (K=3)

Analizując wykresy 45 i 46 możemy zauważyć, że klaster **2** wyróżnia się znacząco od pozostałych klas (patrząc na wykresy pudełkowe i rozkłady gęstości). Natomiast klaster **1** i **2** ma bardzo zbliżone rozkłady. Zatem zmienne `Elong`, `Scat.Ra`, `Pr.Axis.Rect` oraz `Sc.Var.Maxis` mogą być podstawą do odróżnienia obserwacji zawartych w klastrze **2** od pozostałych obserwacji.

2.4.5 Analiza centroidów i medoidów

Tabela 8: Centroidy (k-means) i medoidy (PAM) z rzeczywistymi etykietami (część 1)

Metoda	Klaster	Comp	Circ	D.Circ	Rad.Ra	Pr.Axis.Ra	Max.L.Ra	Scat.Ra	Elong	Pr.Axis.Rect	Max.L.Rect
k-means	1	-0.963	-0.504	-0.911	-1.1	-0.599	-0.488	-0.774	0.84	-0.752	-0.449
k-means	2	1.05	1.07	1.1	0.974	0.178	0.396	1.17	-1.12	1.17	1
k-means	3	-0.286	-0.718	-0.387	-0.0786	0.343	0.00526	-0.582	0.469	-0.606	-0.689
PAM	1	1.61	0.586	1.3	0.949	0.0749	0.675	1.03	-1.07	0.857	0.411
PAM	2	-0.6	-0.857	-0.649	-0.427	0.241	-0.475	-0.718	0.551	-0.634	-0.993
PAM	3	-1.07	-0.376	-1.03	-1.41	-0.924	-0.859	-0.66	0.676	-0.634	-0.291

Tabela 9: Centroidy (k-means) i medoidy (PAM) z rzeczywistymi etykietami (część 2)

Metoda	Klaster	Sc.Var.Maxis	Sc.Var.maxis	Ra.Gyr	Skew.Maxis	Skew.maxis	Kurt.maxis	Kurt.Maxis	Holl.Ra	Etykieta
k-means	1	-0.796	-0.782	-0.369	0.939	-0.131	-0.22	-1.04	-1.04	
k-means	2	1.13	1.17	0.998	-0.0778	0.271	0.182	0.0688	0.181	
k-means	3	-0.517	-0.579	-0.756	-0.756	-0.178	-0.000821	0.852	0.731	
PAM	1	1.16	1.01	1.15	-0.125	0.0166	0.446	0.0874	0.429	3
PAM	2	-0.473	-0.703	-0.955	-0.125	-0.426	-0.287	0.596	0.153	1
PAM	3	-0.756	-0.714	-0.173	1.92	0.0166	-0.165	-1.61	-1.78	1

Tabela 10: Średnie wartości cech (część 1)

Metoda	Klaster	Comp	Circ	D.Circ	Rad.Ra	Pr.Axis.Ra	Max.L.Ra	Scat.Ra	Elong	Pr.Axis.Rect	Max.L.Rect
k-means	1	-0.96	-0.50	-0.91	-1.10	-0.60	-0.49	-0.77	0.84	-0.75	-0.45
k-means	2	1.05	1.07	1.10	0.97	0.18	0.40	1.17	-1.12	1.17	1.00
k-means	3	-0.29	-0.72	-0.39	-0.08	0.34	0.01	-0.58	0.47	-0.61	-0.69
PAM	1	1.05	1.07	1.10	0.97	0.18	0.40	1.17	-1.12	1.17	1.00
PAM	2	-0.39	-0.66	-0.43	-0.22	0.22	-0.06	-0.62	0.53	-0.64	-0.63
PAM	3	-1.04	-0.52	-1.05	-1.24	-0.76	-0.57	-0.77	0.87	-0.75	-0.47

Tabela 11: Średnie wartości cech (część 2)

Metoda	Klaster	Sc.Var.Maxis	Sc.Var.maxis	Ra.Gyr	Skew.Maxis	Skew.maxis	Kurt.maxis	Kurt.Maxis	Holl.Ra
k-means	1	-0.80	-0.78	-0.37	0.94	-0.13	-0.22	-1.04	-1.04
k-means	2	1.13	1.17	1.00	-0.08	0.27	0.18	0.07	0.18
k-means	3	-0.52	-0.58	-0.76	-0.76	-0.18	-0.00	0.85	0.73
PAM	1	1.13	1.17	1.00	-0.08	0.27	0.18	0.07	0.18
PAM	2	-0.55	-0.62	-0.69	-0.56	-0.21	-0.08	0.57	0.48
PAM	3	-0.84	-0.79	-0.34	1.27	-0.05	-0.16	-1.28	-1.29

Analizując centroidy (dla metody k-means) i medoidy (dla metody PAM) z tabel 8-11, można zauważyć, że zmienne Comp, Circ, D.Circ, Rad.Ra i Sc.Var.Maxis, Max.L.Rect dobrze odróżniają klaster *2* (w przypadku k-means). Dla klastra *2* obserwujemy dodatnie wartości, natomiast dla pozostałych klastrów wartości te są ujemne. Zmienne Skew.Maxis, Kurt.Maxis Holl.Ra dla klastra *1* przyjmują natomiast wartości odmienne od klastrów *2* i *3*.

Natomiast w przypadku medoidów zmienne Comp, Circ, D.Circ, Rad.Ra rozróżniają dobrze obserwacje dla klastra *1* od pozostałych.

Różnice te wynikają z innej numeracji klastrów dla metod k-means i PAM.

2.4.6 Wnioski

- Średnia wartość wskaźnika Silhouette dla K=3 wyniosła 0.29, co wskazuje na umiarkowaną jakość grupowania. Metoda ta dobrze separowała klaster odpowiadający klasie “bus”, ale miała problemy z rozróżnieniem klas “opel” i “saab”, które silnie się nakładały.
- Trafność klasyfikacji K-means dla 3 klastrów wyniosła 53.5%, co jest przyzwoitym wynikiem, biorąc pod uwagę trudność separacji danych.
- W przypadku PAM, średnia wartość Silhouette dla K=3 była taka sama jak dla k-means (0.29), ale rozkład wartości wskaźnika był nieco gorszy (więcej obserwacji z ujemnymi wartościami).
- Trafność klasyfikacji dla PAM wyniosła 54.7%, co jest minimalnie lepsze niż k-means.
- Metoda PAM okazała się bardziej odporna na obserwacje odstające niż k-means, ale nadal nie radziła sobie dobrze z nakładającymi się klasami.

- Average linkage. Najlepsza średnia wartość Silhouette (0.36 dla K=3), co mogło sugerować lepszą spójność klastrów niż w przypadku k-means i PAM. Trafność klasyfikacji wyniosła jednak tylko 49.1%, co może wynikać z problemów z przypisaniem obserwacji do odpowiednich klas.
- Complete linkage. Średnia wartość Silhouette wyniosła 0.28, a trafność klasyfikacji 52.2%. Lepiej radziła sobie z równomiernym podziałem danych niż average linkage, ale nadal nie była idealna.
- Single linkage. Najgorsza metoda – średnia wartość Silhouette wyniosła zaledwie 0.08, a trafność klasyfikacji była bardzo niska.
- AGNES (average linkage) – dała najwyższą średnią wartość Silhouette, co wskazuje na najlepszą spójność klastrów.
- Większość metod (zwłaszcza k-means i PAM) lepiej separowała dane przy K=3.
- Wniosek końcowy: W przypadku tego zbioru danych k-means i PAM są lepszym wyborem niż metody hierarchiczne, ale żadna metoda nie osiągnęła bardzo wysokiej skuteczności ze względu na trudną do separacji strukturę danych.