# Data mining - kompleksowy projekt na zbiorze danych Automobile

Damian Lewańczyk

## Spis treści

**1 Analiza eksploracyjna** **2**
1.1 Opis danych . . . 2
1.2 Przypisanie typów zmiennych . . . 4
1.3 Szukanie i wypełnianie brakujących wartości . . . 5
1.4 Wizualizacja danych . . . 6
1.5 Analiza potencjalnych zależności między zmiennymi . . . 13
1.5.1 Analiza graficzna zależności między zmienną objaśnianą, a zmiennymi ilościowymi . . . 13
1.5.2 Analiza graficzna zależności między zmienną objaśnianą, a zmiennymi jakościowymi . . . 16
1.5.3 Analiza formalna zależności między zmienną objaśnianą, a zmiennymi ilościowymi . . . 19
1.5.4 Analiza formalna zależności między zmienną objaśnianą, a zmiennymi jakościowymi . . . 20
1.5.5 Zależności między resztą zmiennych . . . 20

**2 Klasyfikacja zmiennej objaśnianej** **24**
2.1 KNN . . . 25
2.2 Lasy losowe . . . 27
2.3 LDA . . . 31
2.4 QDA . . . 32
2.5 Wielomianowa regresja logistyczna . . . 32
2.6 KDA . . . 33
2.7 Naive Bayes . . . 34

**3 Podsumowanie I części projektu** **35**

**4 Analiza skupień, klasteryzacja** **37**
4.1 Wstęp . . . 37
4.2 Metody grupujace . . . 39
4.2.1 Metoda k -średnich . . . 39
4.3 PAM (partition about medoids) . . . 46
4.4 Algorytm DBSCAN . . . 51
4.5 Metody hierarchiczne . . . 53
4.5.1 AGNES . . . 53
4.5.2 DIANA . . . 65

**5 Redukcja wymiaru** **74**
5.1 PCA - analiza składowych głównych . . . . . 74
5.2 MDS - skalowanie wielowymiarowe . . . . . 83
5.3 Zastosowanie metod redukcji wymiaru przy klasteryzacji . . . . . 88
5.4 Zastosowanie metod redukcji wymiaru przy klasyfikacji . . . . . 93

**6 Podsumowanie drugiej części projektu** **95**

**7 Dodatek** **96**
7.1 AGNES complete linkage - porównania z pozostałymi zmiennymi jakościowymi . 96
7.2 dendrogramy AGNES - wszystkie zmienne . . . . . 99

# 1 Analiza eksploracyjna

## 1.1 Opis danych

Zbiór danych `automobile data set` złożony jest z 205 obserwacji (samochodów). Zawiera 25 różnych zmiennych objaśniających opisujących samochody oraz zmienną objaśnianą `symboling` - zmienną jakościową, która odpowiada poziomowi ryzyka ubezpieczeniowego samochodu, który odpowiada ich cenie. Jeśli samochód jest bardziej niebezpieczny do ubezpieczenia, ten symbol jest korygowany poprzez jego zwiększenie. Wartości są liczbami naturalnym w zakresie od $-3$ do $+3$. Wartość $+3$ wskazuje, że pojazd jest ryzykowny, a $-3$ wskazuje, że jesteśmy najbardziej skłonni go ubezpieczyć. Zanim przejdziemy do analizy danych zapoznamy się ze zmiennymi. Przedstawimy kolejno każdą z nich podając jej nazwę, typ oraz krótki opis:

- **symboling** (zmienna jakościowa, przyjmuje 7 różnych wartości) - określa poziom ryzyka ryzyka ubezpieczeniowego samochodu
- **normalized.losses** (zmienna ilościowa typu ciągłęgo, przyjmuje wartości od 65 do 256) - względna średnia wypłata odszkodowania za ubezpieczony pojazd . Ta wartość jest znormalizowana dla wszystkich samochodów w ramach określonej klasyfikacji (dwudrzwiowe małe, kombi, sportowe/specjalne itp.) i reprezentuje średnią stratę na samochód rocznie
- **make** (zmienna jakościowa, przyjmuje 22 różne wartości) - marka auta
- **fuel.type** (zmienna jakościowa, przyjmuje 2 różne wartości) - typ paliwa dostarczanego silnikowi
- **aspiration** (zmienna jakościowa, przyjmuje 2 różne wartości) - zmienna określająca czy silnik jest wolnossący czy ma doładowanie turbo
- **num.of.doors** (zmienna jakościowa, przyjmuje 2 różne wartości) - określa czy auto jest dwudrzwiowe czy czterodrzwiowe (W Polsce czasem zamiennie odpowiednio 3d i 5D)
- **body.style** (zmienna jakościowa, przyjmuje 5 różnych wartości) - zmienna określająca typ nadwozia
- **drive.wheels** (zmienna jakościowa, przyjmuje 3 różne wartości) - opisuje rodzaj napędu
- **engine.location** (zmienna jakościowa, przyjmuje 2 różne wartości) - zmienna określająca lokalizacje silnika

- **wheel.base** (zmienna ilościowa typu ciągłęgo, przyjmuje wartości od 86.6 do 120.9) - rozstaw osi, czyli odległość między środkami kół poszczególnych osi, mierzona przy symetrycznym ustawieniu kół względem podłużnej osi pojazdu
- **length** (zmienna ilościowa typu ciągłęgo, przyjmuje wartości od 141.1 do 208.1) - długość pojazdu
- **width** (zmienna ilościowa typu ciągłęgo, przyjmuje wartości od 60.3 do 72.3) - szerokość pojazdu
- **heigth** (zmienna ilościowa typu ciągłęgo, przyjmuje wartości od 47.8 do 59.8) - wysokość pojazdu
- **curb.weight** (zmienna ilościowa typu ciągłęgo, przyjmuje wartości od 1488 do 4066) - masa własna pojazdu czyli bez pasażerów, bagażu itd.
- **engine.type** (zmienna jakościowa, przyjmuje 7 różnych wartości) - typ silnika
- **num.of.cylinders** (zmienna jakościowa, przyjmuje 7 różnych wartości) - liczba cylindrów w silniku
- **engine.size** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 61 do 326) - rozmiar silnika.
- fuel.system (zmienna jakościowa, przyjmuje 8 różnych wartości) - rodzaj systemu paliwowego
- **bore** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 2.54 do 3.96) - średnica cylindra
- **stroke** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 2.07 do 4.17) - „Długość skoku", odległość przebyta przez tłok podczas każdego cyklu
- **compression.ratio** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 7 do 23) - stopień kompresji, czyli stosunek objętości cylindra do objętości komory spalania w silniku spalinowym przy ich maksymalnej i minimalnej wartości
- **horsepower** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 48 do 288) - średnica cylindra
- **peak.rpm** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 4150 do 6600) - moment obrotowy silnika
- **city.mpg** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 13 do 49) - zużycie paliwa podczas jazdy w mieście
- **highway.mpg** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 16 do 54) - zużycie paliwa podczas jazdy na autostradzie
- **price** (zmienna ilościowa typu ciągłego, przyjmuje wartości od 5118 do 45400) - wartośc pojazdu

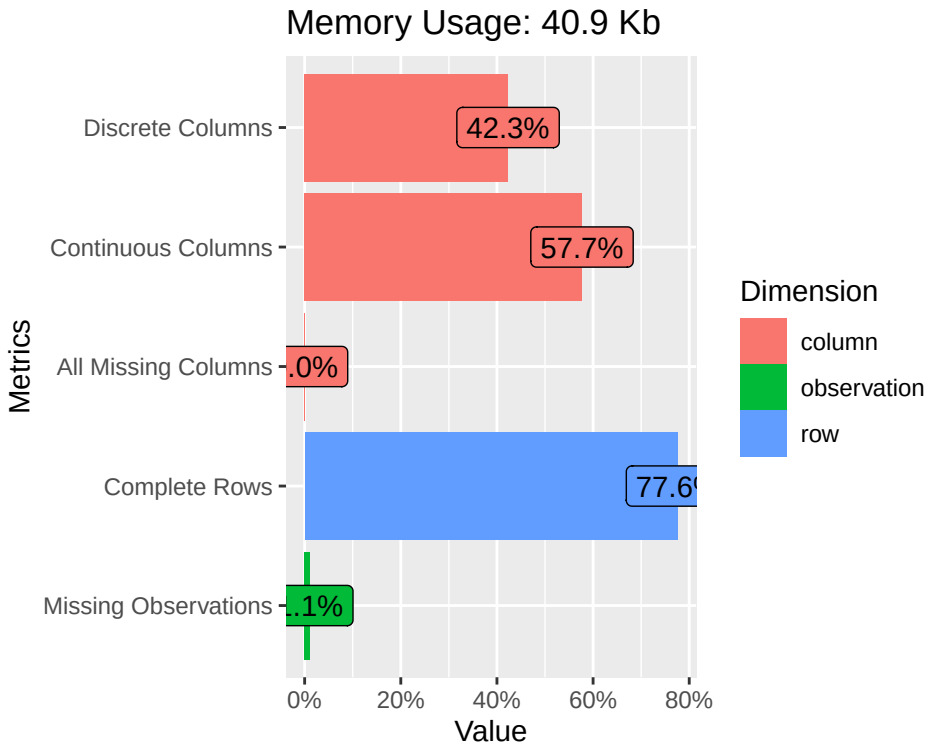
## 1.2 Przypisanie typów zmiennych

Przypiszemy teraz manualnie typy zmiennych jakościowych (factor) i ilościowych (numeric) odpowiednim zmiennym, ponieważ domyślnie R myli się przy przypisywaniu. Następnie spojrzymy wstępnie na kilka początkowych wartości każdej zmiennej z naszego zbioru danych za pomocą funkcji `glimpse` z pakietu **dplyr**.

```
## Rows: 205
## Columns: 26
## $ symboling         <fct> 3, 3, 1, 2, 2, 2, 1, 1, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0~
## $ normalized.losses <int> NA, NA, NA, 164, 164, NA, 158, NA, 158, NA, 192, 192~
## $ make              <fct> alfa-romero, alfa-romero, alfa-romero, audi, audi, a~
## $ fuel.type         <fct> gas, gas, gas, gas, gas, gas, gas, gas, gas, gas, ga~
## $ aspiration        <fct> std, std, std, std, std, std, std, std, turbo, turbo~
## $ num.of.doors      <fct> two, two, two, four, four, two, four, four, four, tw~
## $ body.style        <fct> convertible, convertible, hatchback, sedan, sedan, s~
## $ drive.wheels      <fct> rwd, rwd, rwd, fwd, 4wd, fwd, fwd, fwd, fwd, 4wd, rw~
## $ engine.location   <fct> front, front, front, front, front, front, front, fro~
## $ wheel.base        <dbl> 88.6, 88.6, 94.5, 99.8, 99.4, 99.8, 105.8, 105.8, 10~
## $ length            <dbl> 168.8, 168.8, 171.2, 176.6, 176.6, 177.3, 192.7, 192~
## $ width             <dbl> 64.1, 64.1, 65.5, 66.2, 66.4, 66.3, 71.4, 71.4, 71.4~
## $ height            <dbl> 48.8, 48.8, 52.4, 54.3, 54.3, 53.1, 55.7, 55.7, 55.9~
## $ curb.weight       <int> 2548, 2548, 2823, 2337, 2824, 2507, 2844, 2954, 3086~
## $ engine.type       <fct> dohc, dohc, ohcv, ohc, ohc, ohc, ohc, ohc, ohc, ohc,~
## $ num.of.cylinders  <fct> four, four, six, four, five, five, five, five, five,~
## $ engine.size       <int> 130, 130, 152, 109, 136, 136, 136, 136, 131, 131, 10~
## $ fuel.system       <fct> mpfi, mpfi, mpfi, mpfi, mpfi, mpfi, mpfi, mpfi, mpfi~
## $ bore              <dbl> 3.47, 3.47, 2.68, 3.19, 3.19, 3.19, 3.19, 3.19, 3.13~
## $ stroke            <dbl> 2.68, 2.68, 3.47, 3.40, 3.40, 3.40, 3.40, 3.40, 3.40~
## $ compression.ratio <dbl> 9.00, 9.00, 9.00, 10.00, 8.00, 8.50, 8.50, 8.50, 8.3~
## $ horsepower        <int> 111, 111, 154, 102, 115, 110, 110, 110, 140, 160, 10~
## $ peak.rpm          <int> 5000, 5000, 5000, 5500, 5500, 5500, 5500, 5500, 5500~
## $ city.mpg          <int> 21, 21, 19, 24, 18, 19, 19, 19, 17, 16, 23, 23, 21, ~
## $ highway.mpg       <int> 27, 27, 26, 30, 22, 25, 25, 25, 20, 22, 29, 29, 28, ~
## $ price             <int> 13495, 16500, 16500, 13950, 17450, 15250, 17710, 189~
```

Następnie dokonamy wstępnego rozeznania w danych za pomocą funkcji `introduce` i `plot intro` z pakietu **DataExplorer**.

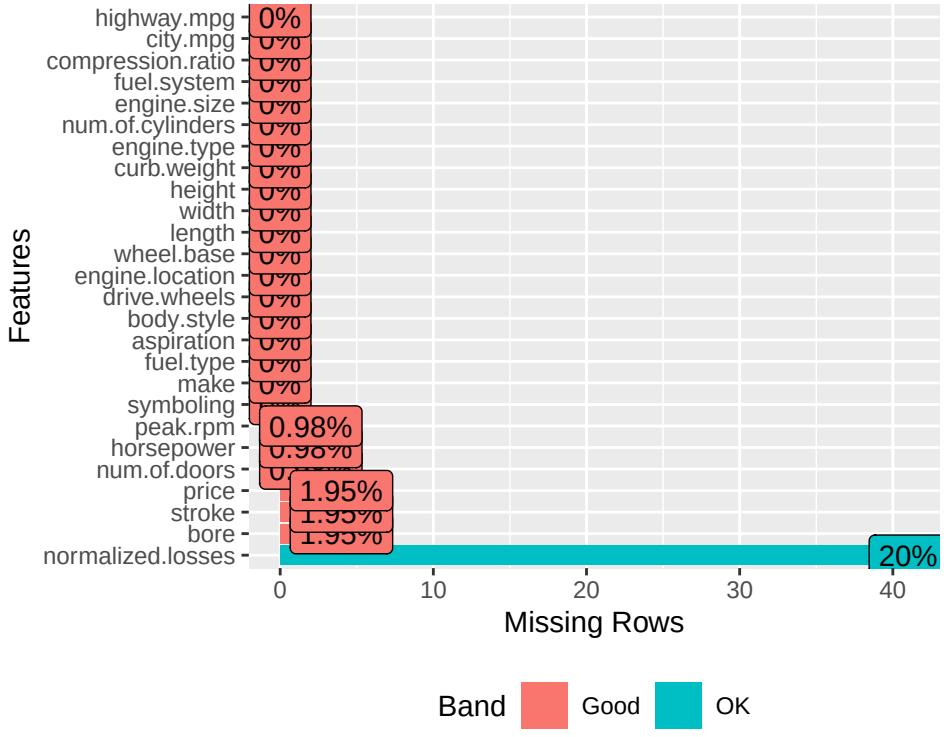
```
##                        [,1]
## rows                    205
## columns                  26
## discrete_columns         11
## continuous_columns       15
## all_missing_columns       0
## total_missing_values     59
## complete_rows           159
## total_observations     5330
## memory_usage          41832
```




Jak widzimy, odpowiednio przypisaliśmy typy zmiennych w **R**. Jak możemy zauważyć, w naszych danych są wartości brakujące, którymi zajmiemy się nimi w następnym podrozdziale.

## 1.3 Szukanie i wypełnianie brakujących wartości

Na początek, użyjemy funkcji `plot missing` z pakietu **DataExplorer**, aby sprawdzić jakie zmienne oraz jaki ich odsetek ma brakujące wartości.




Możemy też sprawdzić nie względną, a liczbową wartość wartości brakujących dla każdej

kolumny:

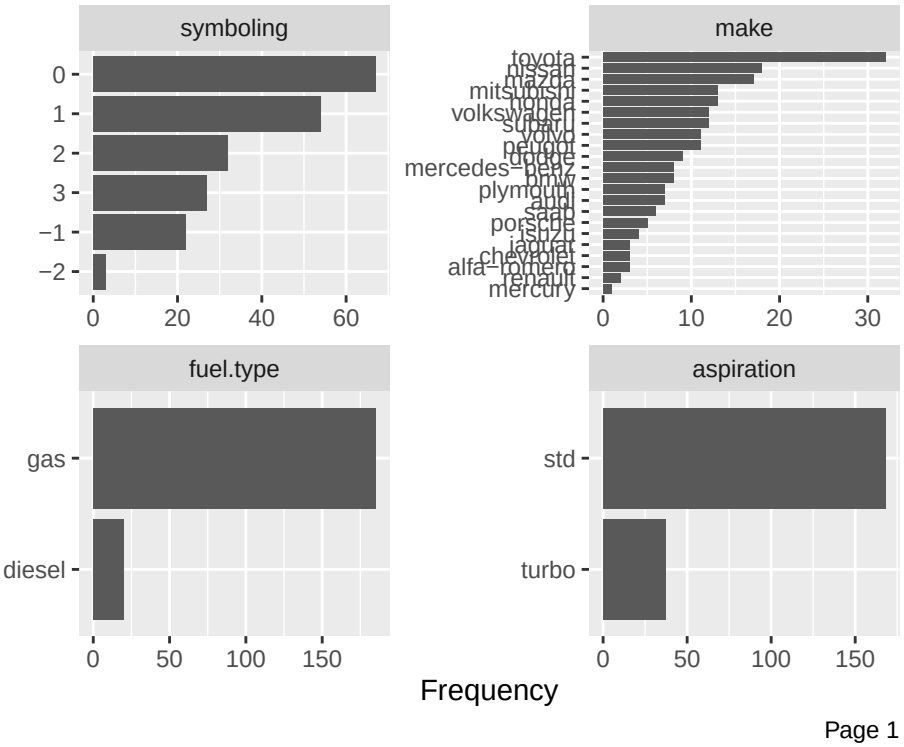
```
apply(data,2,function(x){length(which(is.na(x)))})
```

```
##         symboling normalized.losses              make         fuel.type
##                 0                41                 0                 0
##        aspiration      num.of.doors        body.style      drive.wheels
##                 0                 2                 0                 0
##   engine.location        wheel.base            length             width
##                 0                 0                 0                 0
##            height       curb.weight       engine.type  num.of.cylinders
##                 0                 0                 0                 0
##       engine.size       fuel.system              bore            stroke
##                 0                 0                 4                 4
## compression.ratio        horsepower          peak.rpm          city.mpg
##                 0                 2                 2                 0
##       highway.mpg             price
##                 0                 4
```

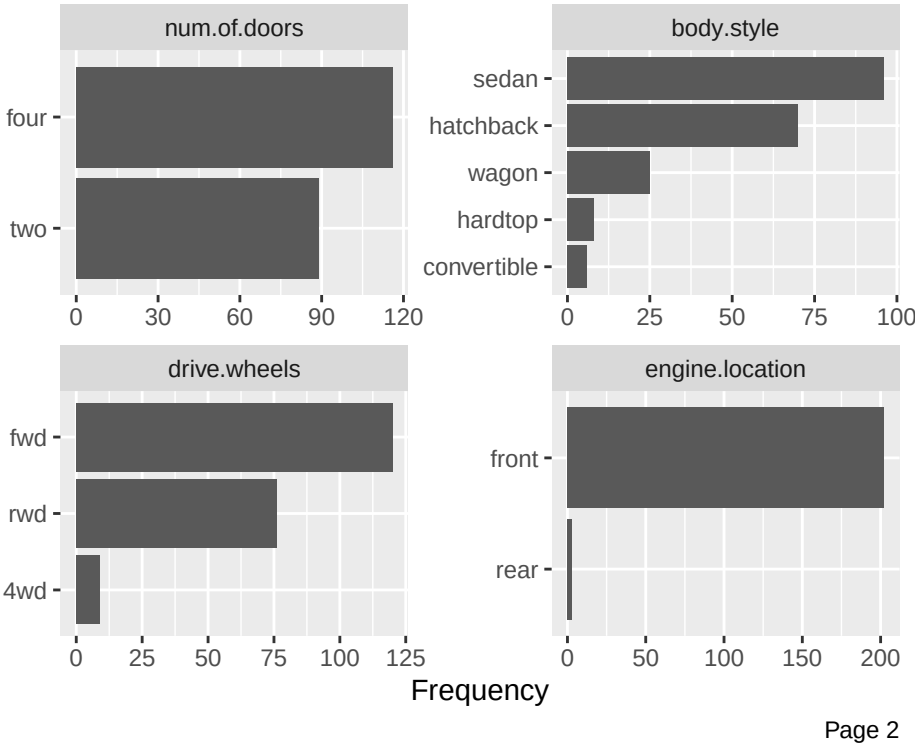
Zamieniamy wartości $NA$ na konkretne liczby tj. na średnie ze zmiennych ilościowych typu ciągłego, a wartości $NA$ zmiennej jakościowej `num.of.doors` zamienamy na wartości najczęściej występujące, w tym wypadku będzie to wartość `four`.

## 1.4 Wizualizacja danych

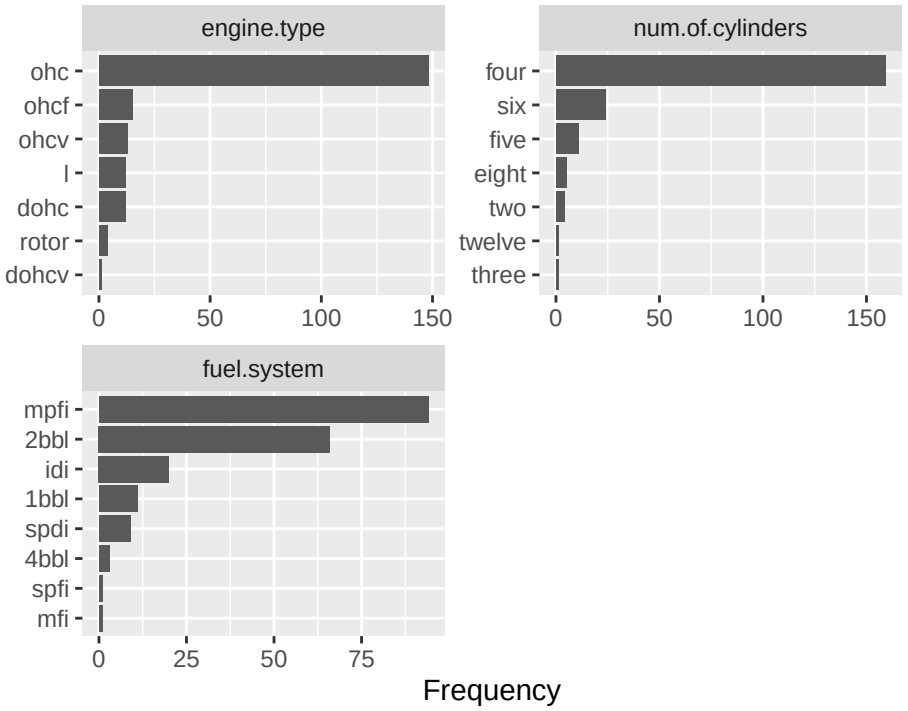
W tym podrozdziale zajmiemy się wizualizacją danych. Na początek zaprezentowane zostaną barploty dla zmiennych jakościowych, które przedstawią częstości występowania poszczególnych wartości dla każdej zmiennej typu factor. Zrobimy to za pomocą funkcji `plot bar` z pakietu **DataExplorer**. Widzimy je na rysunkach 1, 2 i 3 poniżej.




Rysunek 1: Częstotliwości występowania kategorii dla zmiennych jakościowych




Rysunek 2: Częstotliwości występowania kategorii dla zmiennych jakościowych

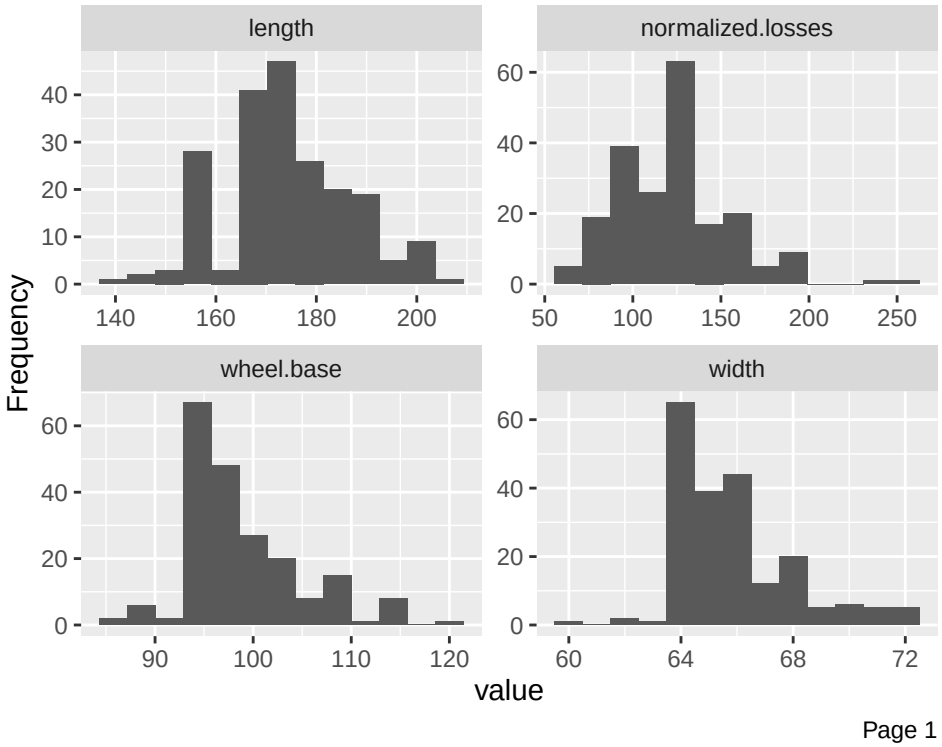



Rysunek 3: Częstotliwości występowania kategorii dla zmiennych jakościowych

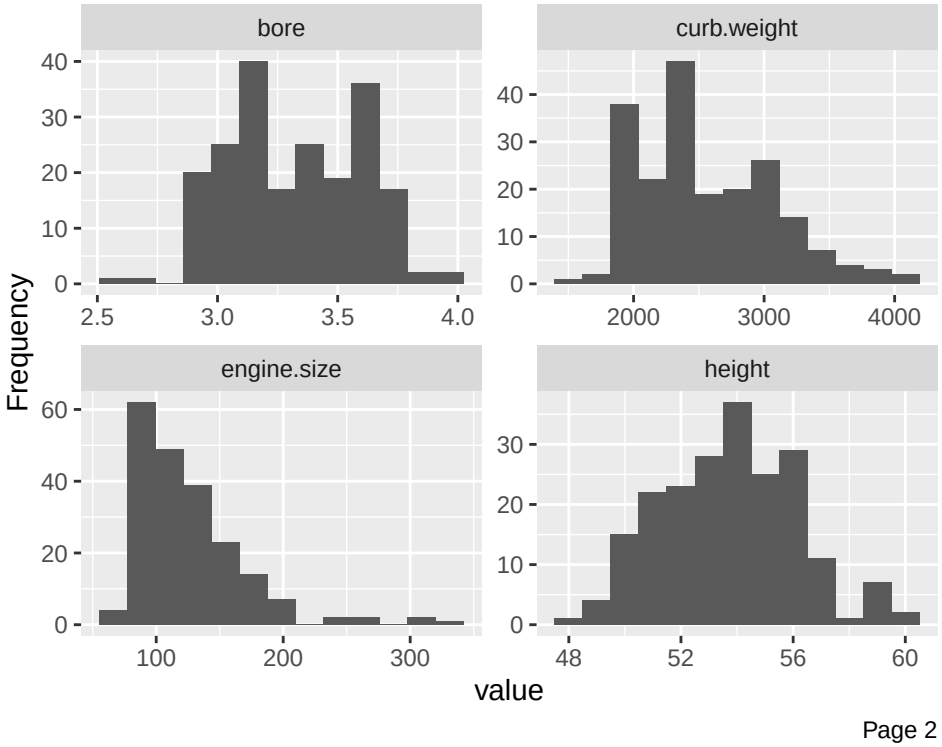
Na podstawie powyższych wykresów, z modelu usuwamy dwie zmienne: `fuel.type` oraz `engine.location`, ponieważ występuje w nich za duża dysproporcja między występowaniem klas, przez co nie będą one przekazywać nam praktycznie żadnych informacji. Drugą ważną informacją jest fakt, że nasza jakościowa zmienna objaśniana w ogólnie nie przyjmuje jednej z potencjalnych wartości tj. $-3$, a kolejną wartość tj. $-2$ przyjmuje zaledwie kilka razy. Poza tym, szczególną uwagę trzeba zwrócić na fakt, że jedna ze zmiennych jakościowych - `make` ma bardzo dużo kategorii (ponad 20), co może być problemem przy ewentualnych próbach transformacji takiej zmiennej na numeryczną (np. za pomoca One-hot encoding), przez znaczące zwiększenie wymiaru)

Dodatkowo, jest kilka innych zmiennych: `normalized.losses`, `engine.type`, `drive.wheels`, `fuel.type` i `aspiration`, które są potencjalnymi kandydatami do usunięcia z modelu. Zmienna `normalized.losses`, bo ma duży odsetek wartości brakujących (które na razie zostały zastąpione wartościami średnimi), a reszta zmiennych - która jest typu jakościowego - z powodu dominacji jednej wartości zmiennej. Ostateczną decyzję podejmiemy po dalszej analizie, czy owe zmienne mogą mieć znaczący wpływ na naszą zmienną objaśnianą `symboling`.

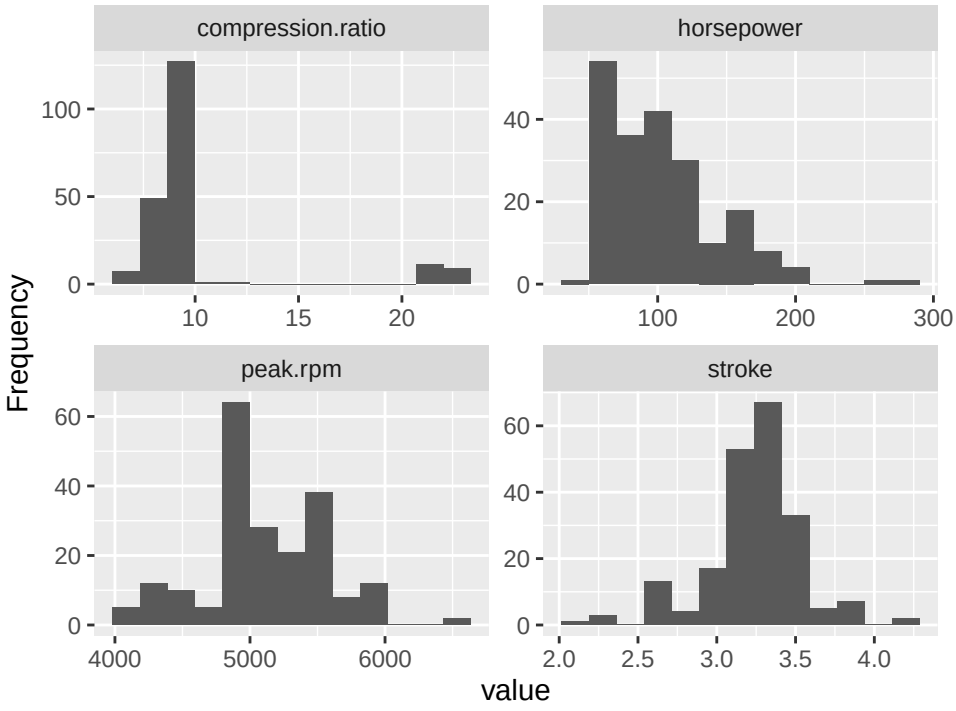
Poniżej, na rysunkach 4, 5, 6 i 7 przedstawione zostały histogramy dla zmiennych ilościowych. Wygenerowane zostały one za pomocą funkcji `plot histogram` z pakietu **DataExplorer**.



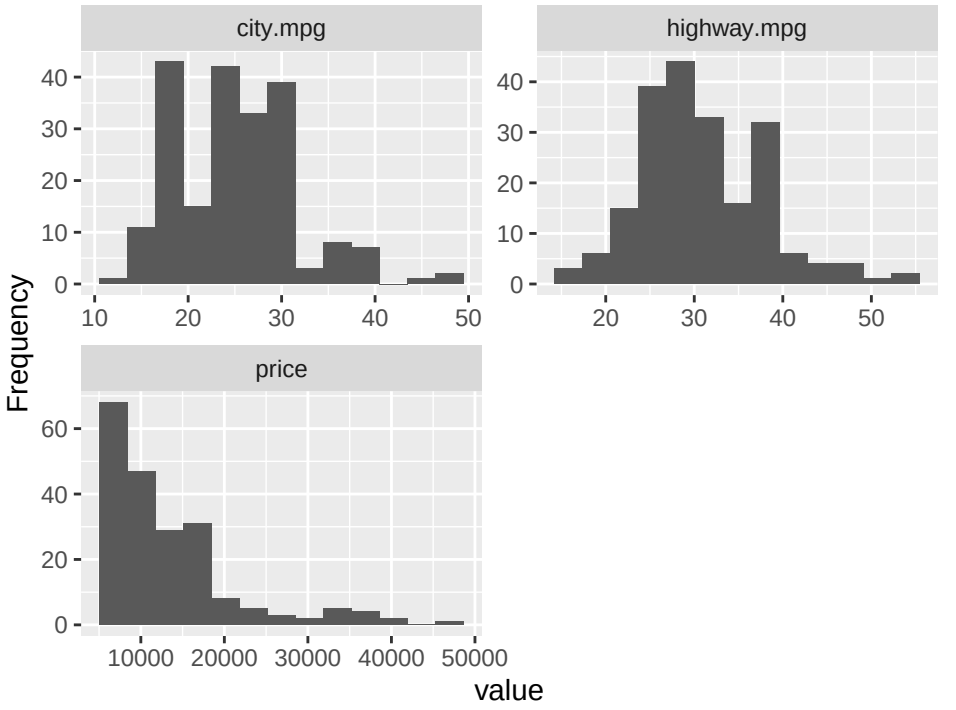

Rysunek 4: Histogramy dla zmiennych ilościowych




Rysunek 5: Histogramy dla zmiennych ilościowych




Rysunek 6: Histogramy dla zmiennych ilościowych

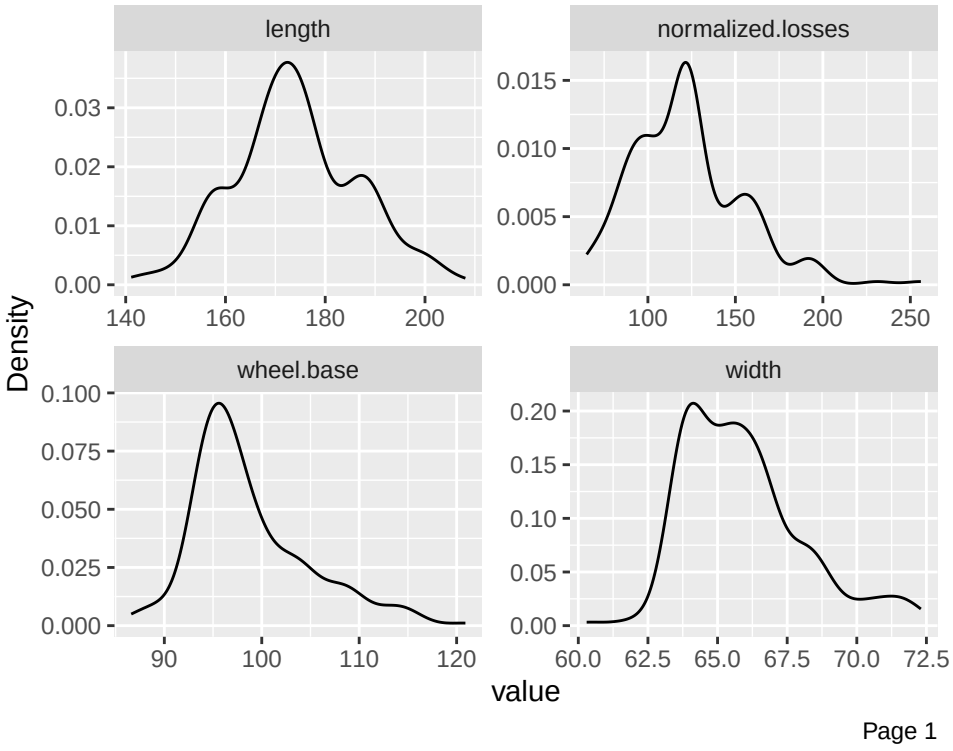



Rysunek 7: Histogramy dla zmiennych ilościowych

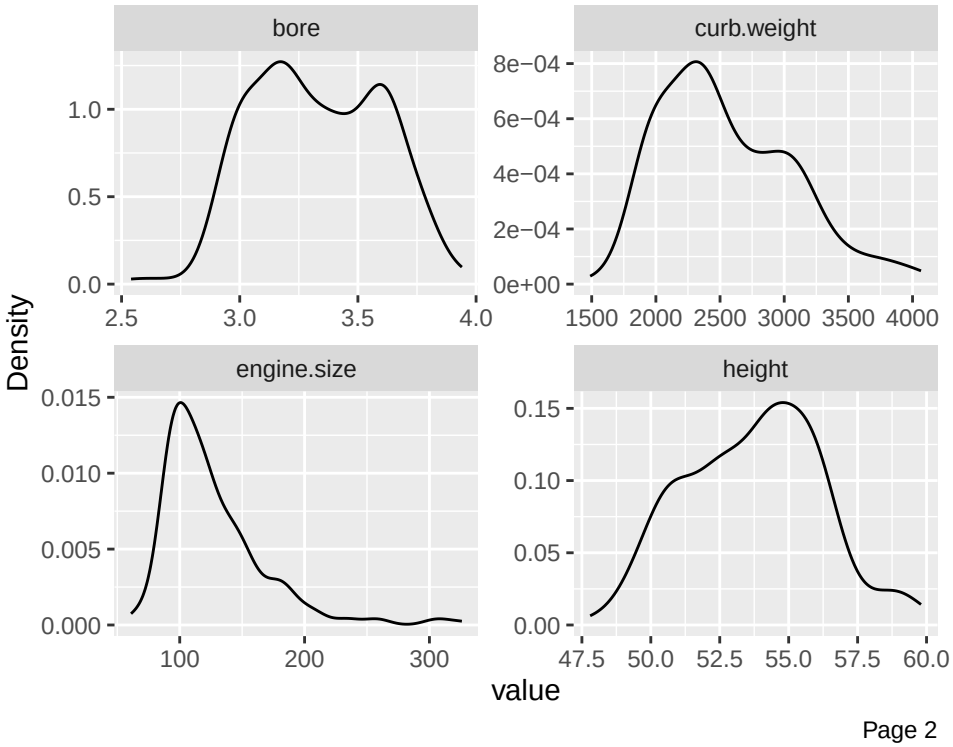
Jak możemy zobaczyć, rozkłady zmiennych ilościowych znacząco różnią się kształtem między sobą. I tak, widzimy że np. zmienne `length`, `bore` czy `stroke` mogą pochodzić z roz-

kładu bardziej przypominającego rozkład normalny, natomiast rozkłady takich zmiennych jak `horsepower`, `price` czy `engine.size` bardziej przypominają np. rozkład wykładniczy.

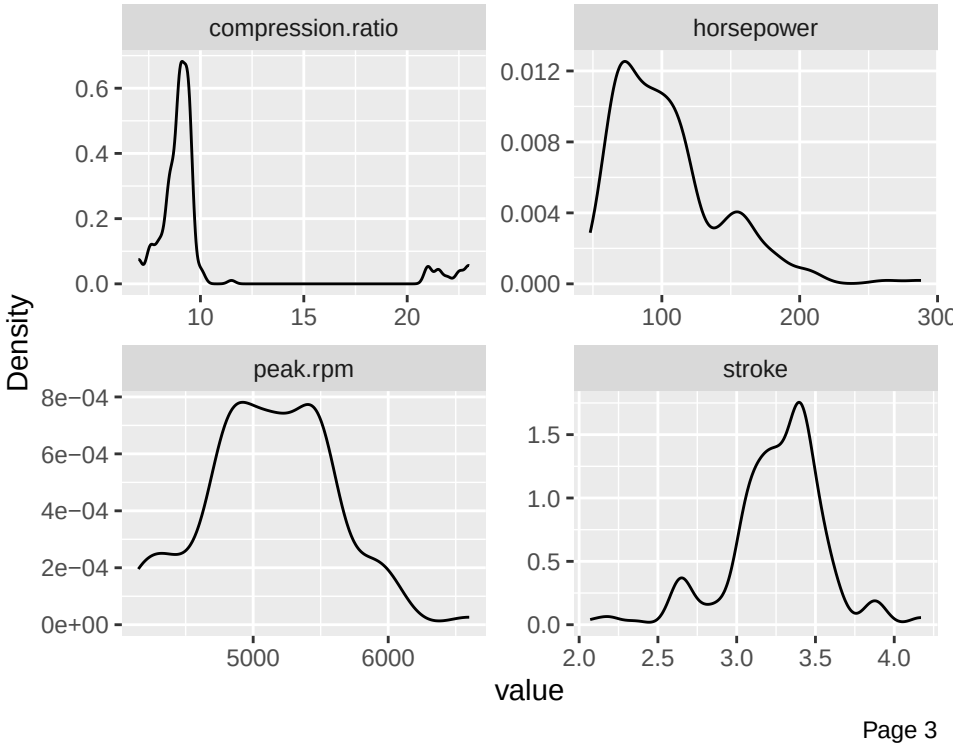
Poniżej, na rysunkach 8, 9, 10 i 11 dodatkowo wygnerowane zostały estymowane gęstości dla zmiennych ilościowych. Stworzone zostały przy pomocy funkcji `plot density` z pakietu **DataExplorer**. Wykresy te potwierdzają nasze obserwacje co do różnych charakterystyk zmiennych numerycznych.



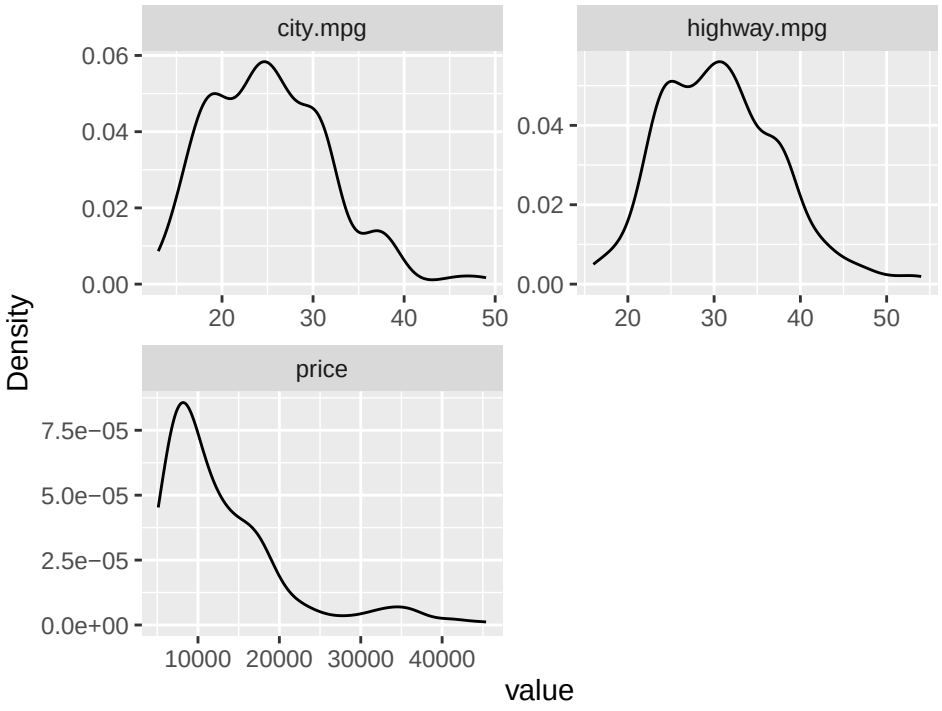

Rysunek 8: Estymowane gęstości dla zmiennych ilościowych




Rysunek 9: Estymowane gęstości dla zmiennych ilościowych




Rysunek 10: Estymowane gęstości dla zmiennych ilościowych



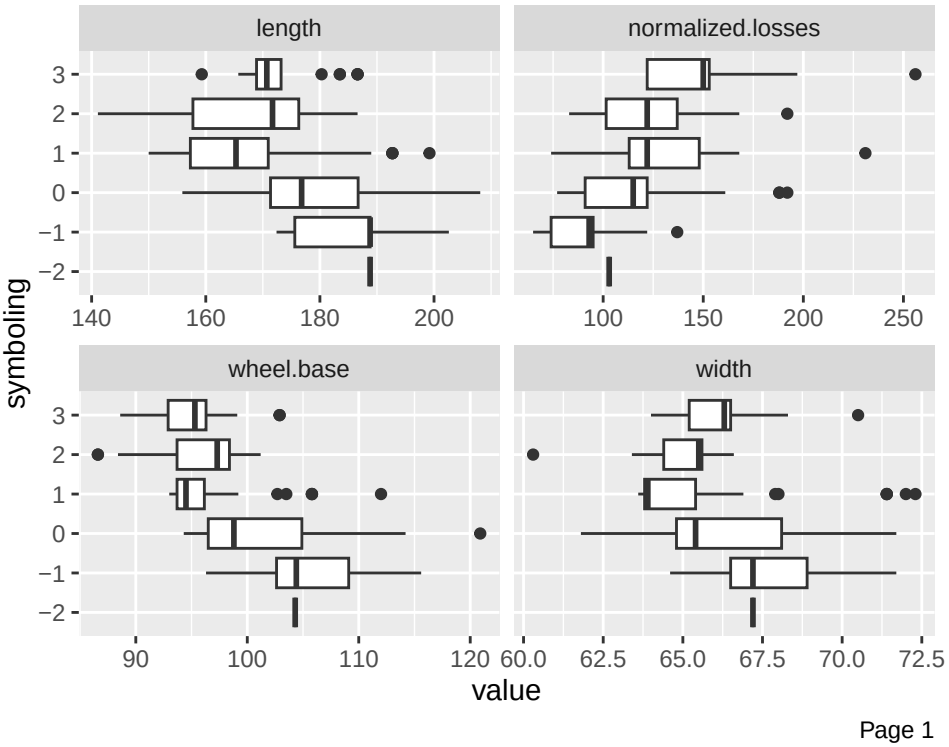

Rysunek 11: Estymowane gęstości dla zmiennych ilościowych

## 1.5 Analiza potencjalnych zależności między zmiennymi

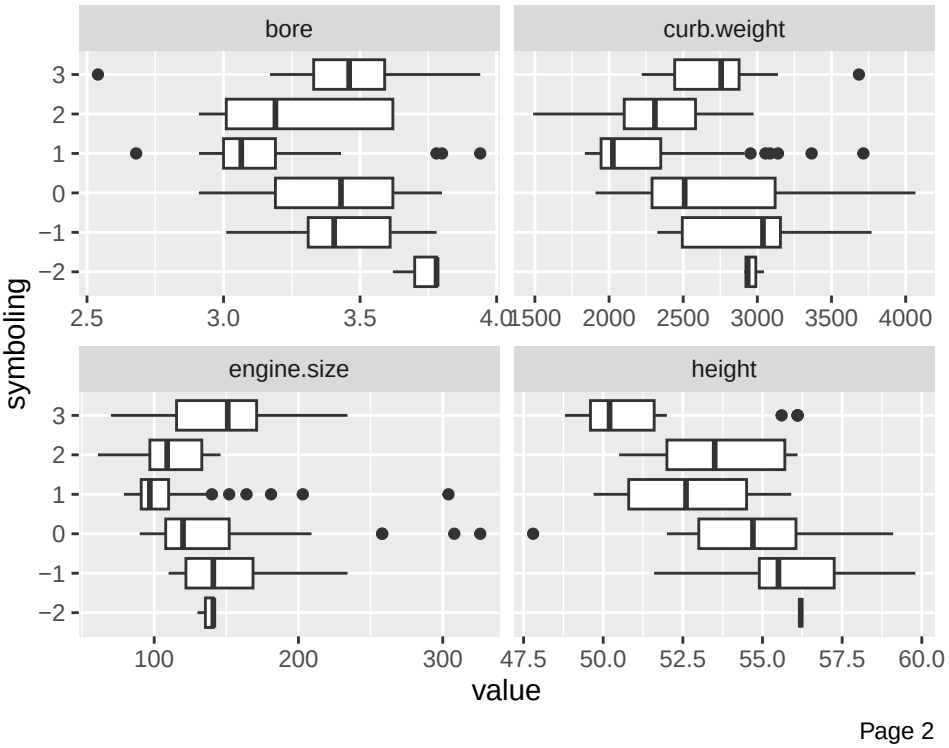
### 1.5.1 Analiza graficzna zależności między zmienną objaśnianą, a zmiennymi ilościowymi

Oczywiście najbardziej interesują nas potencjalne zależności między zmienną objaśnianą `symboling`, a pozostałymi zmiennymi. Dlatego głównie zajmiemy się właśnie nimi, na początek przedstawiając wykresy "pogrupowane" na kategorie wartości zmiennej `symboling`: wykresy pudełkowe dla zmiennych ilościowych oraz wykresy mozaikowe dla zmiennych jakościowych.

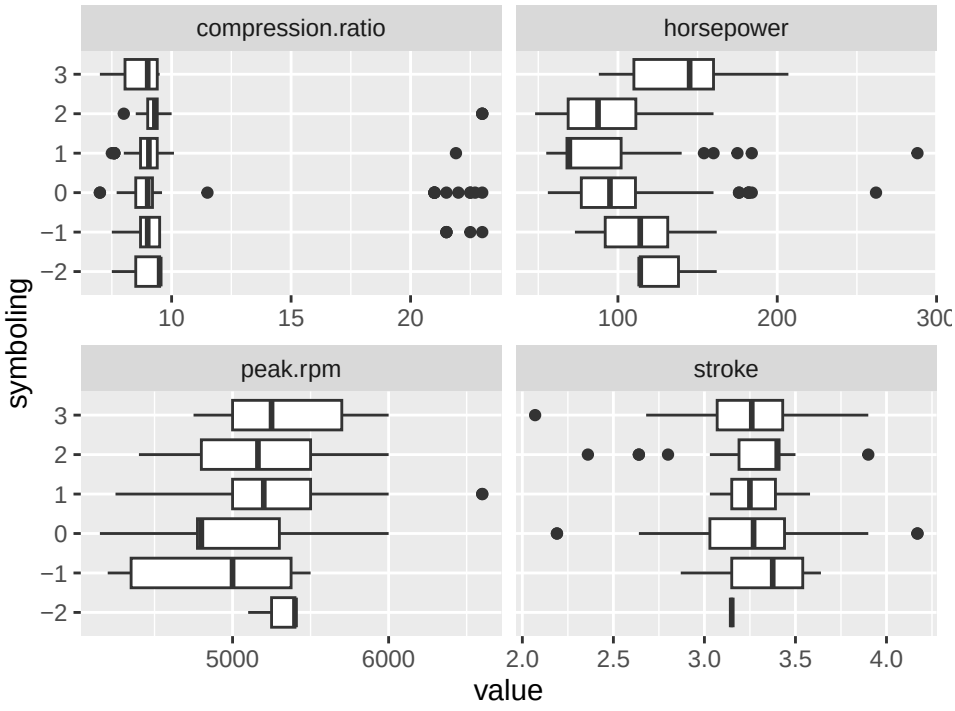
Poniżej, na rysunkach 12, 13, 14 i 15 przedstawione są wykresy pudełkowe dla zmiennych ilościowych pogrupowane wg. wartości zmiennej `symboling`. Robimy je przy użyciu funkcji `plot boxplot` z pakietu **DataExplorer**.



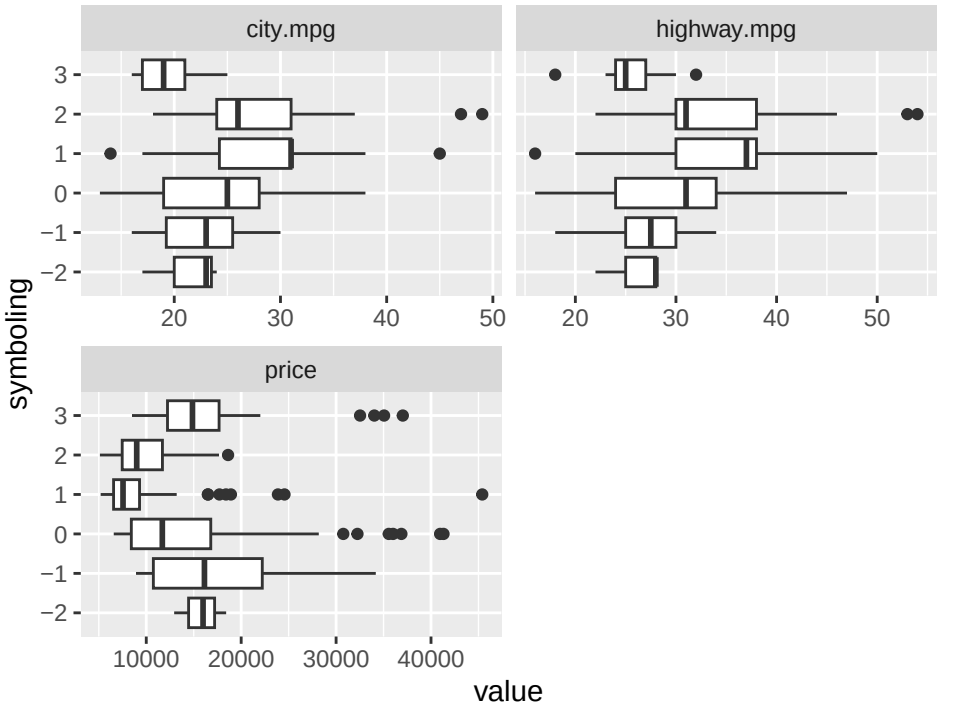

Rysunek 12: Wykresy pudełkowe dla zmiennych ilościowych ze względu na wartości zmiennej symboling




Rysunek 13: Wykresy pudełkowe dla zmiennych ilościowych ze względu na wartości zmiennej symboling




Rysunek 14: Wykresy pudełkowe dla zmiennych ilościowych ze względu na wartości zmiennej symboling



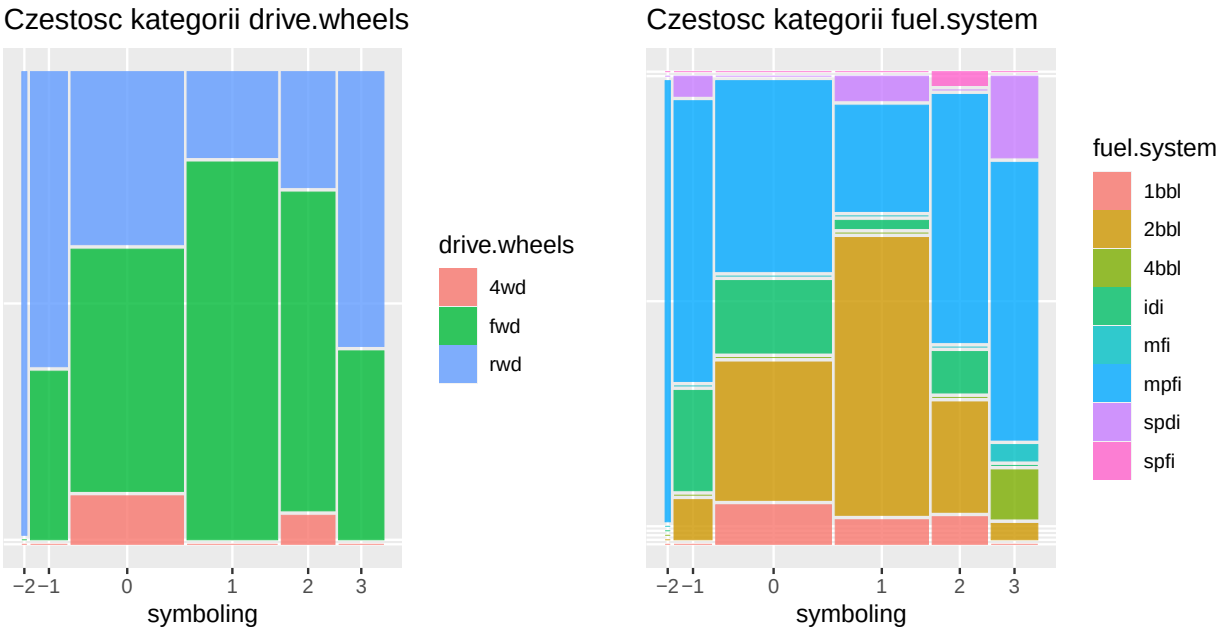

Rysunek 15: Wykresy pudełkowe dla zmiennych ilościowych ze względu na wartości zmiennej symboling

Obserwując wykres powyżej, możemy stwierdzić, że praktycznie dla każdej zmiennej numerycznej wykresy pudełkowe dla poszczególnych wartości zmiennej objaśnianej `symboling` różnią się od siebie. Najbardziej podobne są dla zmiennej `compression.ratio`, `peak.rpm` i `stroke`. Widzimy też na podstawie tych wykresów, że zależności między zmiennymi objaśniającymi, a zmienną `symboling` są różne. Jednakże najbliżej zależności, którą nazwalibyśmy liniową (biorąc pod uwagę, że zmienna objaśniana ma kategorie liczbowe i jest porządkowa) są zmienne: `length`, `normalized.losses`, `wheel.base` i `height`. Zależności między zmienną objaśnianą, a zmiennymi `length`, `wheel.base` i `height` określającymi fizyczne parametry auta są blizko uznania za "malejące", tzn. wraz ze wzrostem któregokolwiek z tych parametrów, zmienna symboling przyjmuje statystycznie mniejsze wartości czyli auto jest mniej ryzykowne. Natomiast odwrotnie jest w przypadku zmiennej `normalized.losses`, czyli wartości wypłat za pojazd, gdzie im większe jej wartości, tym większa wartość zmiennej `symboling`, czyli auto jest potencjalnie bardziej ryzykowne.

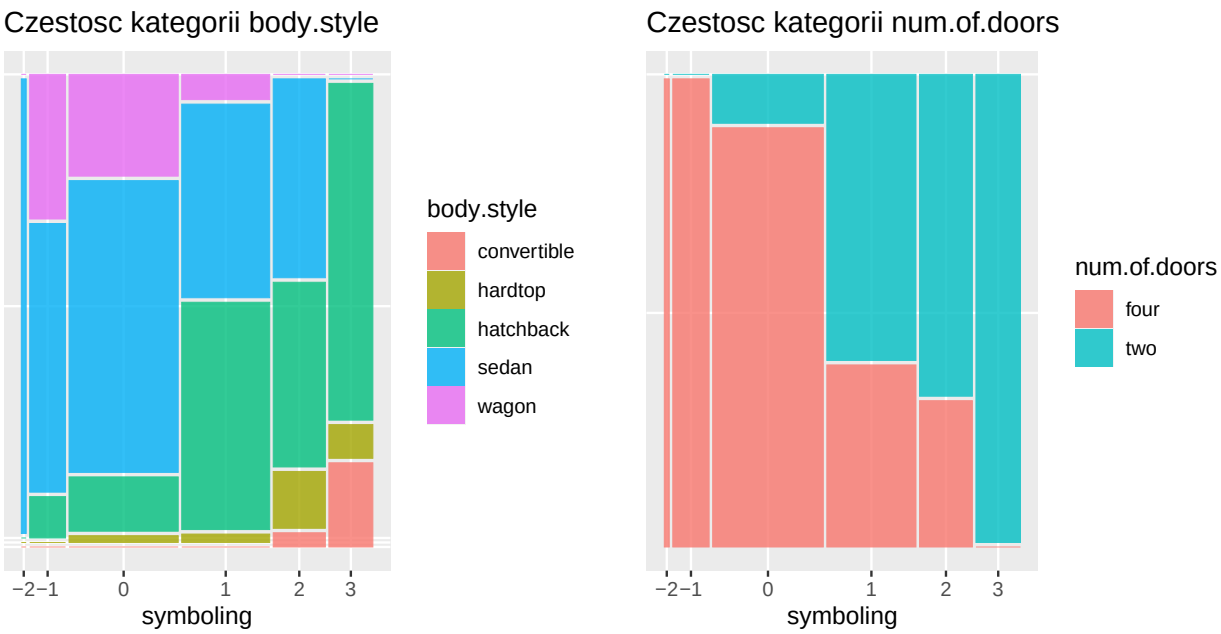
Na koniec dodajmy, że z racji tego, że zmienna `normalized.losses` jest tak istotna, oczywiście nie usuwamy jej z modelu, a wartości brakujące zostają z podmienionymi za nie średnimi z reszty wartości.

### 1.5.2 Analiza graficzna zależności między zmienną objaśnianą, a zmiennymi jakościowymi

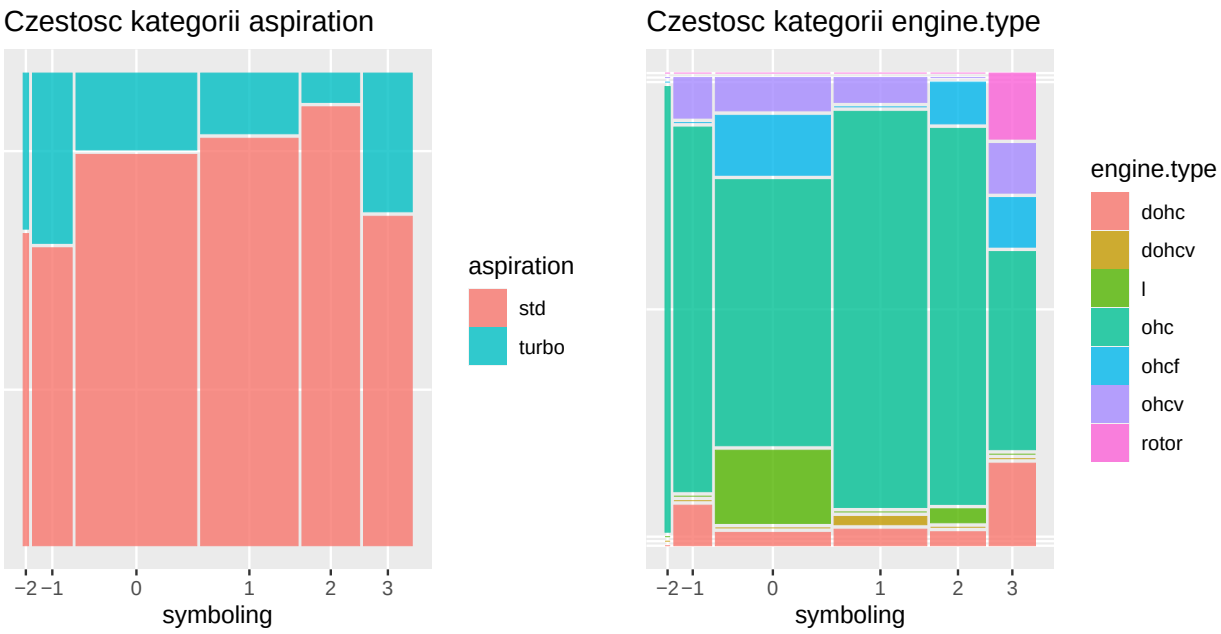
Poniżej, na rysunkach 16, 17, 18 i 19 przedstawione są wykresy mozaikowe dla zmiennych jakościowch pogrupowane wg. wartości zmiennej `symboling`.




Rysunek 16: Częstotliwość występowania poszczególnych kategorii zmiennych jakościowych ze względu na wartości zmiennej symboling

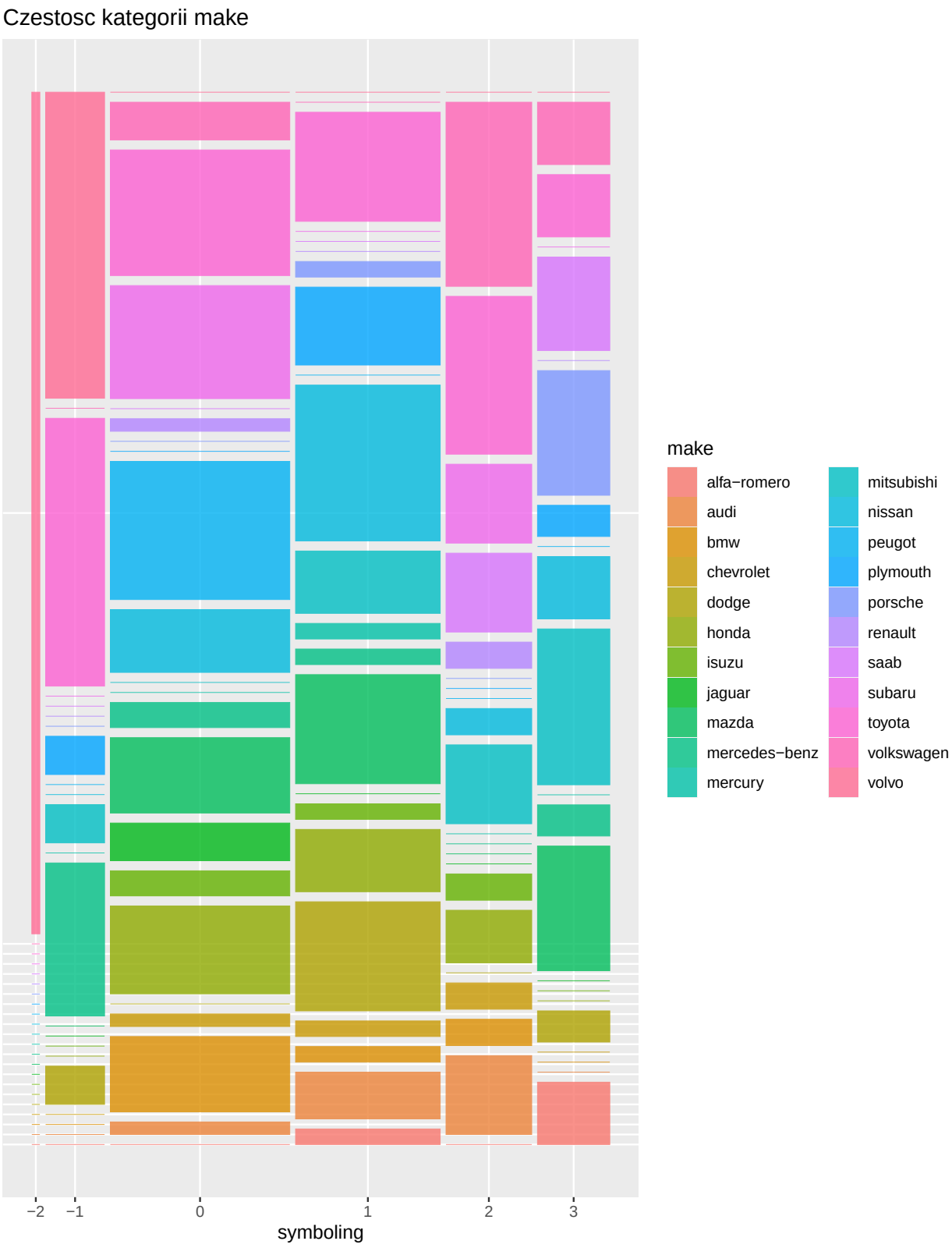



Rysunek 17: Częstotliwość występowania poszczególnych kategorii zmiennych jakościowych ze względu na wartości zmiennej symboling




Rysunek 18: Częstotliwość występowania poszczególnych kategorii zmiennych jakościowych ze względu na wartości zmiennej symboling

Warto zauważyć, że kolory na legendzie są "do góry nogami" względem tego jak poszczególne kategorie przedstawione są na wykresach. Jest tak na wszystkich wykresach mozaikowych zaprezentowanych tutaj, ale warto na to zwrócić uwagę szczególnie przy następnym, dotyczącym mark aut:




Rysunek 19: Częstotliwość występowania poszczególnych kategorii zmiennych jakościowych ze względu na wartości zmiennej symboling

Na powyższych wykresach jasno widać, że każda zmienna jakościowa ma mniejszy lub większy wpływ na zmienną `symboling`. Wydaje się bez głębszej analizy, że najmniejszy wpływ ma zmienna `aspiration`. Zdecydowanie najprostszą do interpretacji jest relacja tej zmiennej ze

zmienną `num.of.doors`. Widzimy, że jeśli auto jest dwudrzwiowe, to wzrasta ryzyko jego ubezpieczenia względem auta czterodrzwiowego. Innym bardzo ciekawym, choć bardzo złożonym wykresem, jest ten ze zmienną `make`, w przypadku której liczba kategorii znacząco utrudnia opis tego wykresu. Możemy zauważyć, bardzo różne rozkłady kategorii dla poszczególnych wartości zmiennej `symboling`. Np. wszystkie wartości $-2$ zmiennej `symboling` są dla marki auta volvo, a reszta volvo ma wartość `symboling` równą $-1$, natomiast zdecydowana większość aut marki alfa romeo znajduje się w wysokiej grupie ryzyka (`symboling`=3).

### 1.5.3 Analiza formalna zależności między zmienną objaśnianą, a zmiennymi ilościowymi

Teraz przejdziemy do bardziej formalnej części analizy zależności. Dla zmiennych ilościowych. tabeli 1 przedstawione zostały wartości odpowiednio $\eta^2$, $\omega^2$ oraz wartości p-value dla testu Kruskala - Walisa. $\eta^2$, $\omega^2$ są typami wielkości efektu [6] między zmienną ilościową, a jakościową. Wartości tych współczynników w **R** zostały obliczone z pomocą funkcji `anova` oraz `lm` z pakietu **stats** [4][3]. Istnieje wiele różnych interpretacji tych współczynników (od jakiej wartości zmienną możemy uznać za istotną/średnio istotną/ bardzo istotną) - więcej tutaj [7], [8]. Nam jednakże tak samo bardzo zależy po prostu na uszeregowaniu zmiennych od tych najbardziej istotnych względem zmiennej objaśnianej, do tych najmniej istotnych. Tak samo jeśli chodzi o wartości p-value dla testu Kruskala-Walisa. Wartości te w **R** zostały obliczone za pomocą funkcji `kruskal.test` z pakietu **stats** [5]. Test ten nie zakłada normalności rozkładów. Hipotezą zerową jest równość dystrybuant rozkładów w porównywanych populacjach (w naszym wypadku, dla różnych wartości zmiennej `symboling` tzn. im mniejsza p-value tym teoretycznie większa szansa na to, że dystrybuanty rozkładów są różne dla poszczególnych klas, czyli większa szansa na to, że zmienna ilościowa jest w pewien sposób zależna od jakościowej (i na odwrót). Więcej tutaj [9]. W tabeli poniżej zmienne zostały uszeregowane

| | Zmienna ilościowa | $\eta$ | omega squared | kruskal p value |
|---|---|---|---|---|
| 11 | wheel.base | 0.360 | 0.253 | 3.82E-16 |
| 1 | height | 0.350 | 0.247 | 2.29E-13 |
| 15 | length | 0.271 | 0.199 | 1.81E-11 |
| 2 | normalized.losses | 0.260 | 0.192 | 1.33E-10 |
| 8 | curb.weight | 0.249 | 0.184 | 8.80E-12 |
| 12 | bore | 0.235 | 0.175 | 3.95E-10 |
| 6 | highway.mpg | 0.218 | 0.163 | 5.29E-10 |
| 5 | city.mpg | 0.214 | 0.160 | 1.60E-09 |
| 3 | horsepower | 0.173 | 0.130 | 3.36E-09 |
| 10 | width | 0.158 | 0.118 | 4.71E-09 |
| 7 | price | 0.147 | 0.110 | 1.00E-10 |
| 4 | peak.rpm | 0.131 | 0.096 | 1.12E-04 |
| 9 | engine.size | 0.114 | 0.082 | 1.07E-08 |
| 13 | compression.ratio | 0.065 | 0.039 | 1.64E-01 |
| 14 | stroke | 0.017 | -0.008 | 2.92E-01 |

Tabela 1: Porównanie współczynników dla zmiennych ilościowych

Jak możemy zauważyc w tabeli powyżej, hierarchia zmiennych jest dokładnie taka sama dla współczynników $\eta^2$, $\omega^2$ oraz bardzo podobna dla wartości p-value (tylko, że tam rosnąco

zamiast malejąco). Co więcej możemy stwierdzić, że wyniki w tabeli prowadzą do bardzo podobnych wniosków co analiza graficzna jeśli chodzi o analizę hierarchii zależności między zmiennymi ilościowymi, a zmienną `symboling` tzn. te same zmienne (`wheel.base`, `height`, `length`, `normalized.losses`) są w grupie najbardziej istotnych i te same zmienne (`compression.ratio`, `stroke`) są w grupie najmniej istotnych.

### 1.5.4 Analiza formalna zależności między zmienną objaśnianą, a zmiennymi jakościowymi

Natomiast dla porównania miary związku między jakościową zmienną objaśnianą, a innymi zmiennymi jakosciowymi użyliśmy współczynnika V Craméra [1]. W tym celu zastosowaliśmy funkcję `cramerV` z pakietu **rcompanion** [2]. Współczynnik ten przyjmuje wartości od 0 do 1, przy czym im większa jego wartość, tym większy związek między poszczególnymi zmiennymi. Współczynniki zostały uszeregowane od największego i przedstawione w tabeli 2.

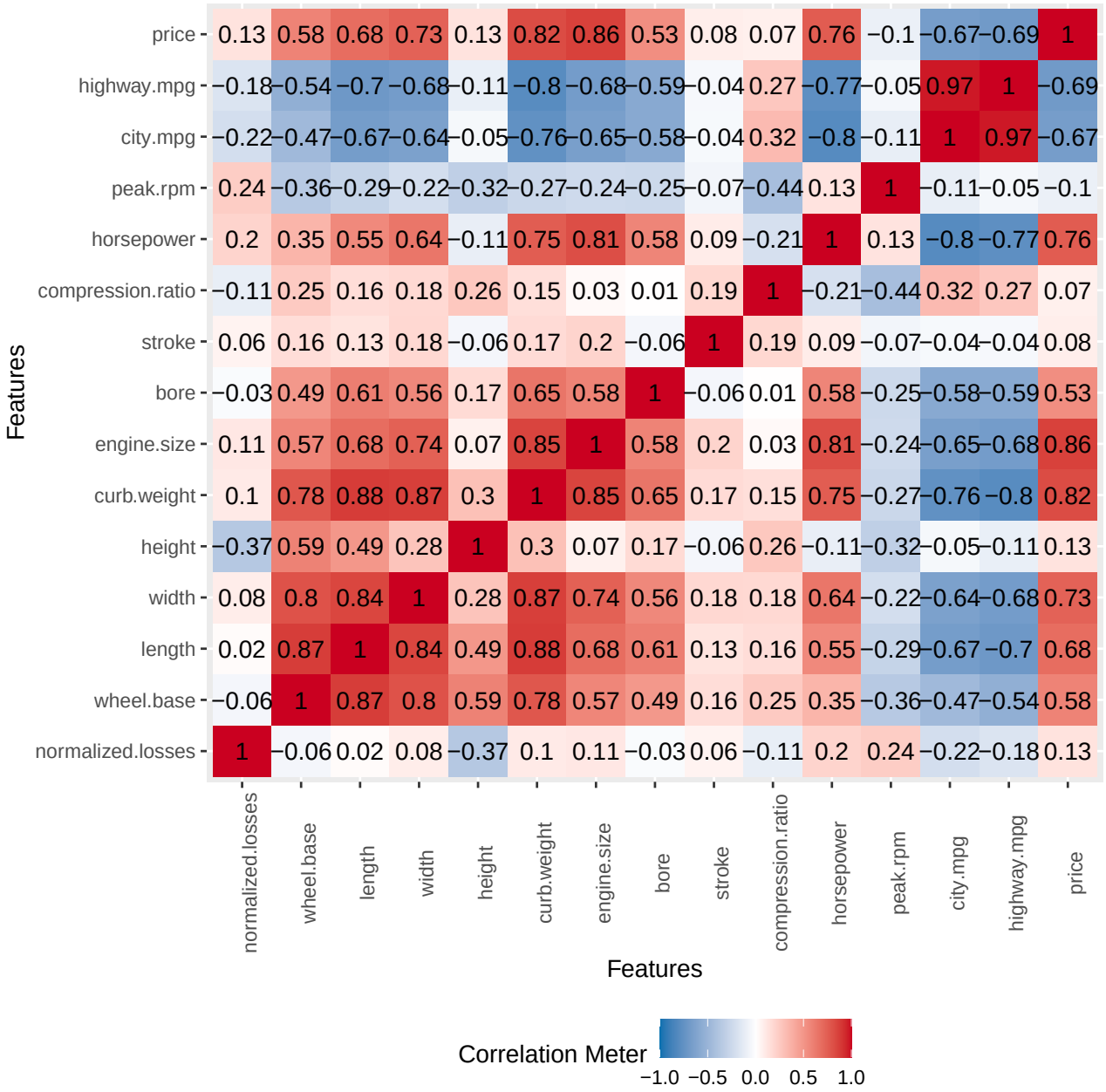
| | Zmienna jakościowa | Współczynnik V Cramera |
|---|---|---|
| 4 | num.of.doors | 0.696 |
| 1 | make | 0.542 |
| 2 | body.style | 0.366 |
| 8 | fuel.system | 0.321 |
| 5 | drive.wheels | 0.307 |
| 6 | engine.type | 0.279 |
| 3 | aspiration | 0.242 |
| 7 | num.of.cylinders | 0.233 |

Tabela 2: Wartości V Cram

Jak możemy zauważyć powyżej, również w przypadku zmiennych jakościowych analiza formalna prowadzi do podobnych wniosków co analiza graficzna, mianowicie zmienna `num.of.doors` faktycznie ma zdecydowanie największą wartość współczynnika, natomiast druga w tej hierarchii zmienna `make` również ma współczynnik zdecydowanie większy od pozostałych. Dodatkowo możemy zaobserwować, że przypuszczenie o tym, że zmienna `aspiration` jest mało istotna jest potwierdzone przez jej współczynnik V Cramera, i tego jak blisko najmniejszej wartości się znajduje.

### 1.5.5 Zależności między resztą zmiennych

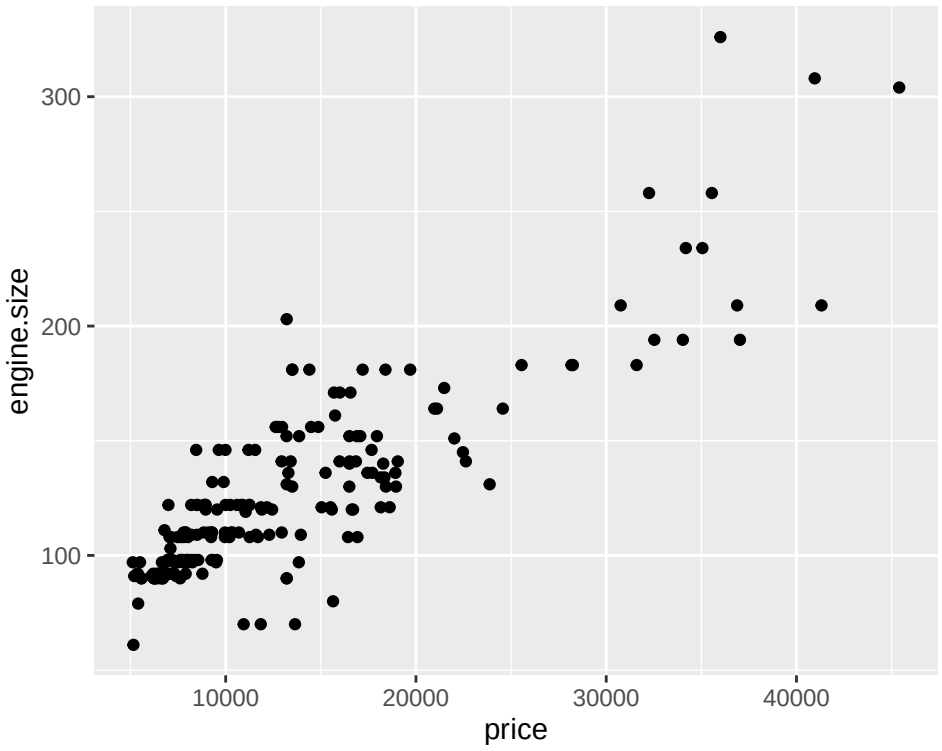
Poniżej, na rysunku 20 zamieszczona zotała tablica korelacji między wszystkimi zmiennymi ilościowymi z mapą ciepła. Stworzona została ona przy pomocy funkcji `plot corelation` z pakietu **DataExplorer**.



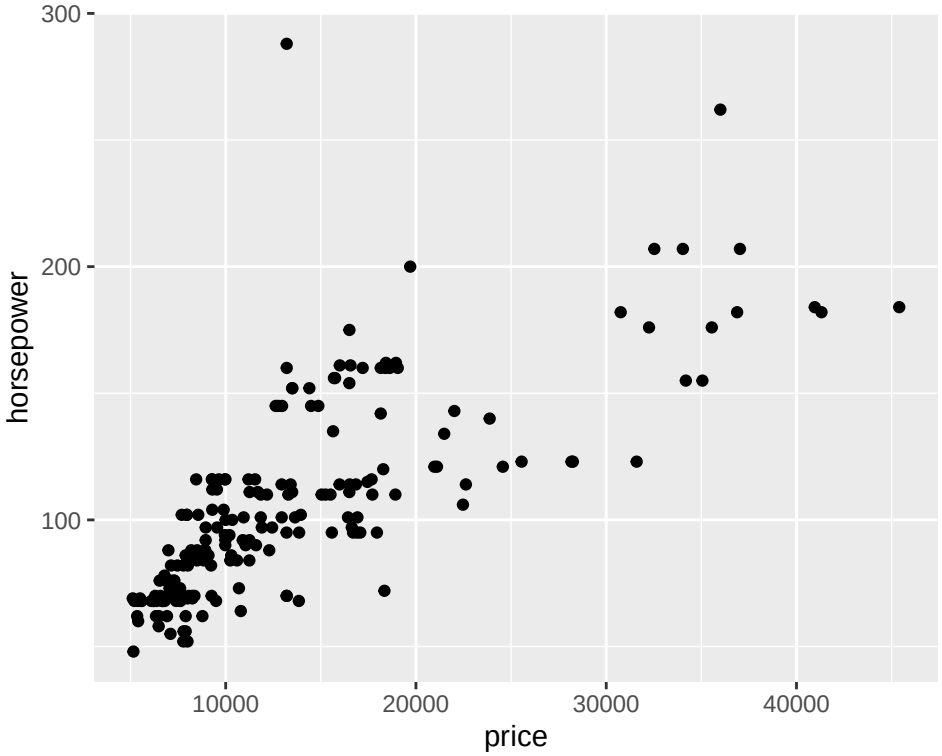

Rysunek 20: Tabela korelacji między zmiennymi ilościowymi

Jak widzimy powyżej, między zmiennymi ilościowymi istnieje bardzo dużo silnych korelacji, najsilniej skorelowane są zmienne dotyczące spalania paliwa w mieście i na autostradzie, dodatkowo widzimy dużo korelacji o wartości bezwzględnej przekraczającej 0.8, a jeszcze więcej wartości większych od 0.7. Ponadto możemy stwierdzić, że najmniej skorelowanymi z resztą zmiennych są `compression.ratio`, `stroke` i `normalized.losses`

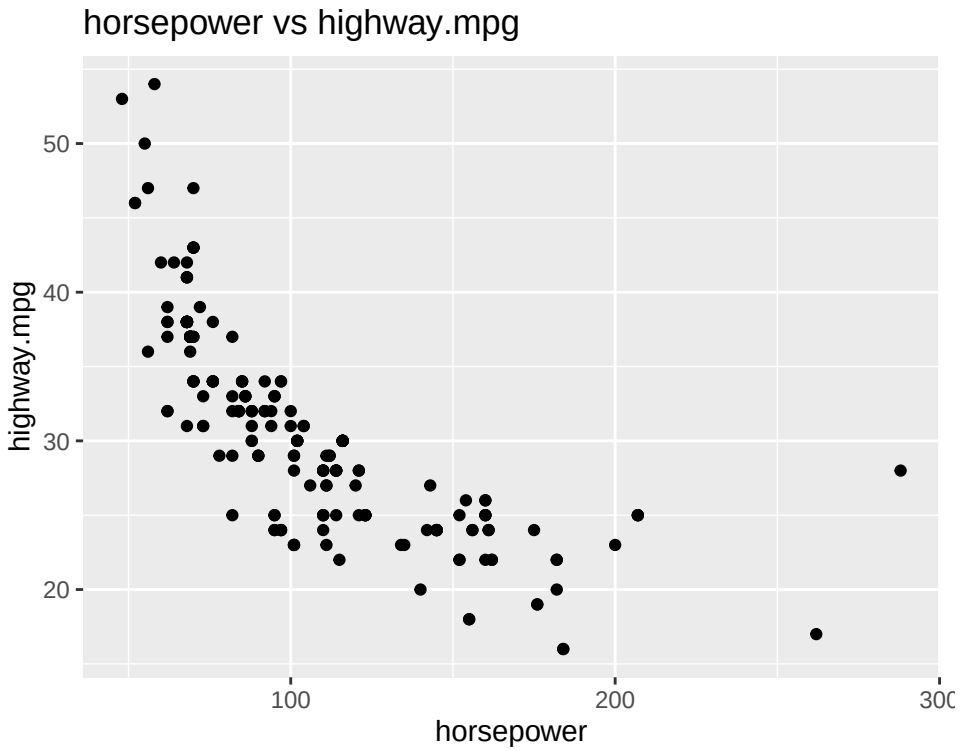
Ponadto stworzone zostało kilka scatterplotów między różnymi parami zmiennych ilościowych. Zostały one przedstawione na rysunkach 21, 22, 23 i 24.



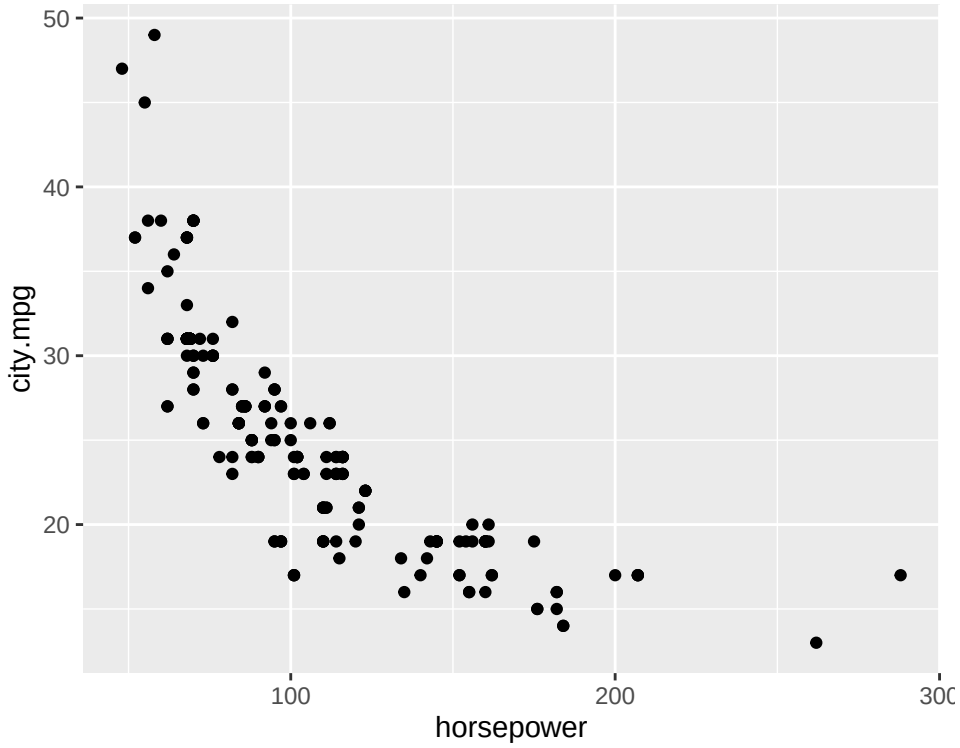

Rysunek 21: Porównanie zależności między zmiennymi engine.size i price




Rysunek 22: Porównanie zależności między zmiennymi horsepower i price




Rysunek 23: Porównanie zależności między zmiennymi ehorsepower i highway.mpg

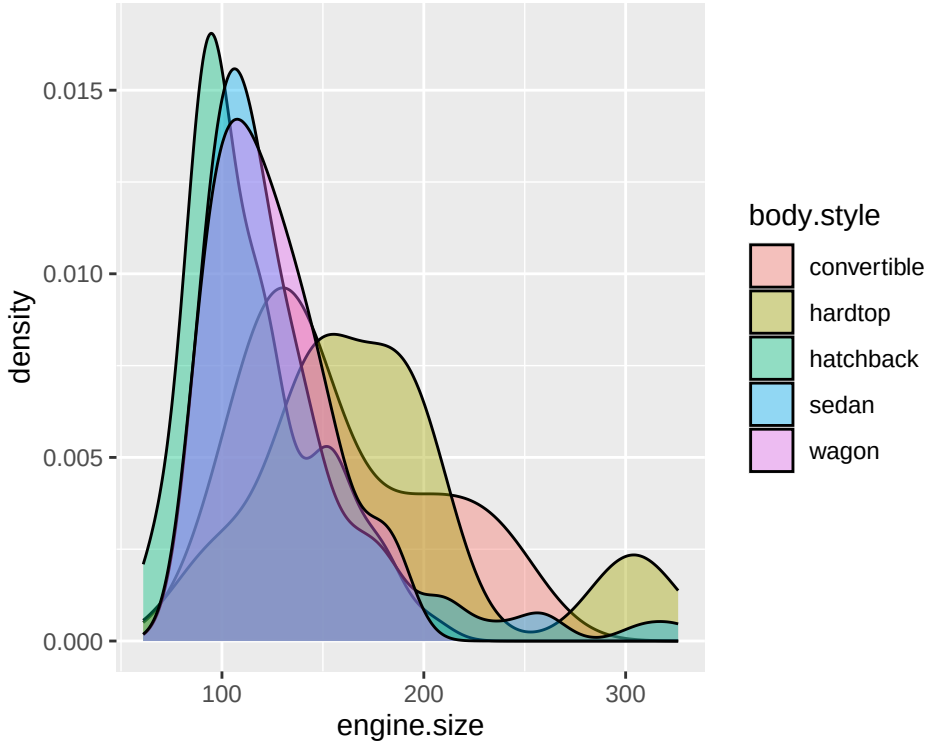



Rysunek 24: Porównanie zależności między zmiennymi engine.size i city.mpg

Na podstawie wykresów powyżej, możemy potwierdzić kilka wartości liczbowych z tablicy korelacji 20. Dwa pierwsze wykresy, czyli 21, 22 przedstawiające odpowiednio zależności między

ceną, a rozmiarem silnika i ceną, a mocą silnika potwiewrdzają obliczone wartości korelacji - 0.86 i 0.76 odpowiednio. Widzimy, że faktycznie wraz ze wzrostem ceny wartości obu tych zmiennych mają tendencję rosnącą. Natomiast jeśli chodzi o wykresy 23 i 24, przedstawiające zależności między liczbą koni mechanicznych a spalaniem paliwa przez auto na autostradzie oraz w mieście, one również potwierdzają obliczone wartości korelacji, które wynosiły odpowiednio $-0.77$ i $-0.8$. Widzimy, że wraz ze wzrostem liczby koni mechannicznych oba typy spalania mają tendencję spadkową.

Poniżej, na rysunku 25 przedstawiony został wykres porównujący estymowane gęstości zmiennej ilościowej `engine.size` ze względu na różne wartości zmiennej jakościowej `body.style`. Wykres ten pokazuje, że istnieją różnice dla rozkładów rozmiaru silnika w zależności od rodzaju nadwozia.




Rysunek 25: Porównanie gęstości zmiennej engine.size ze względu na różne wartości zmiennej body.style

# 2 Klasyfikacja zmiennej objaśnianej

Na poczatek, stworzymy podzbiór naszych danych złożony tylko ze zmiennych ilościowych oraz zmiennej objaśnianej `symboling`. Tworzymy go na potrzeby niektórych modeli, które nie mogą przyjmować zmiennych jakościowych.

```
data_num <- data[c("symboling", int_kol, num_kol)]
```

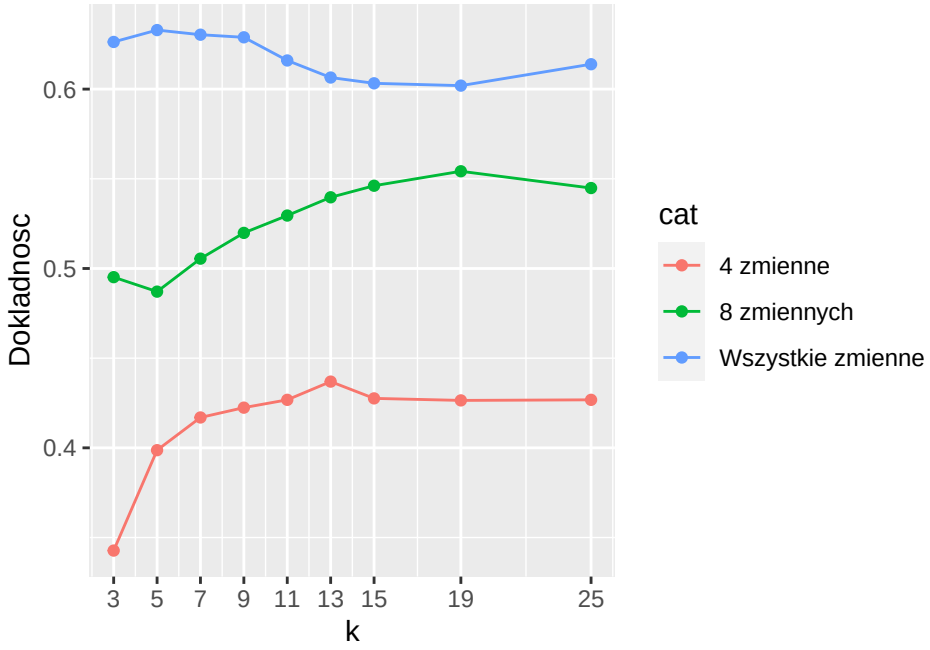
## 2.1 KNN

Na samym starcie zaimplementujemy metodę k-najbliższych sąsiadów dla $k = 3$, $k = 5$, $k = 7$, $k = 9$, $k = 11$, $k = 13$, $k = 15$, $k = 19$, $k = 25$ oraz dla następujących podzbiorów zmiennych:

- wszystkie zmienne numeryczne
- 8 wybranych najbardziej istotnych zmiennych numerycznych: `normalized.losses`, `height`, `length`, `wheel.base`, `curb.weight`, `bore`, `city.mpg` i `highway.mpg`
- 4 wybrane najbardziej istotne zmienne numeryczne: `normalized.losses`, `height`, `length`, `wheel.base`

Do tego celu użyjemy funkcji `ipredknn` z pakietu **ipred**. Zbiór treningowy będzie obejmował 0.7 rozmiaru całej próbki, naszą metryką będzie dokładność, a całą procedurę powtórzymy 100 razy, żeby uśrednić otrzymane rezultaty, a także sprawdzić rozrzut otrzymanych rezultatów.

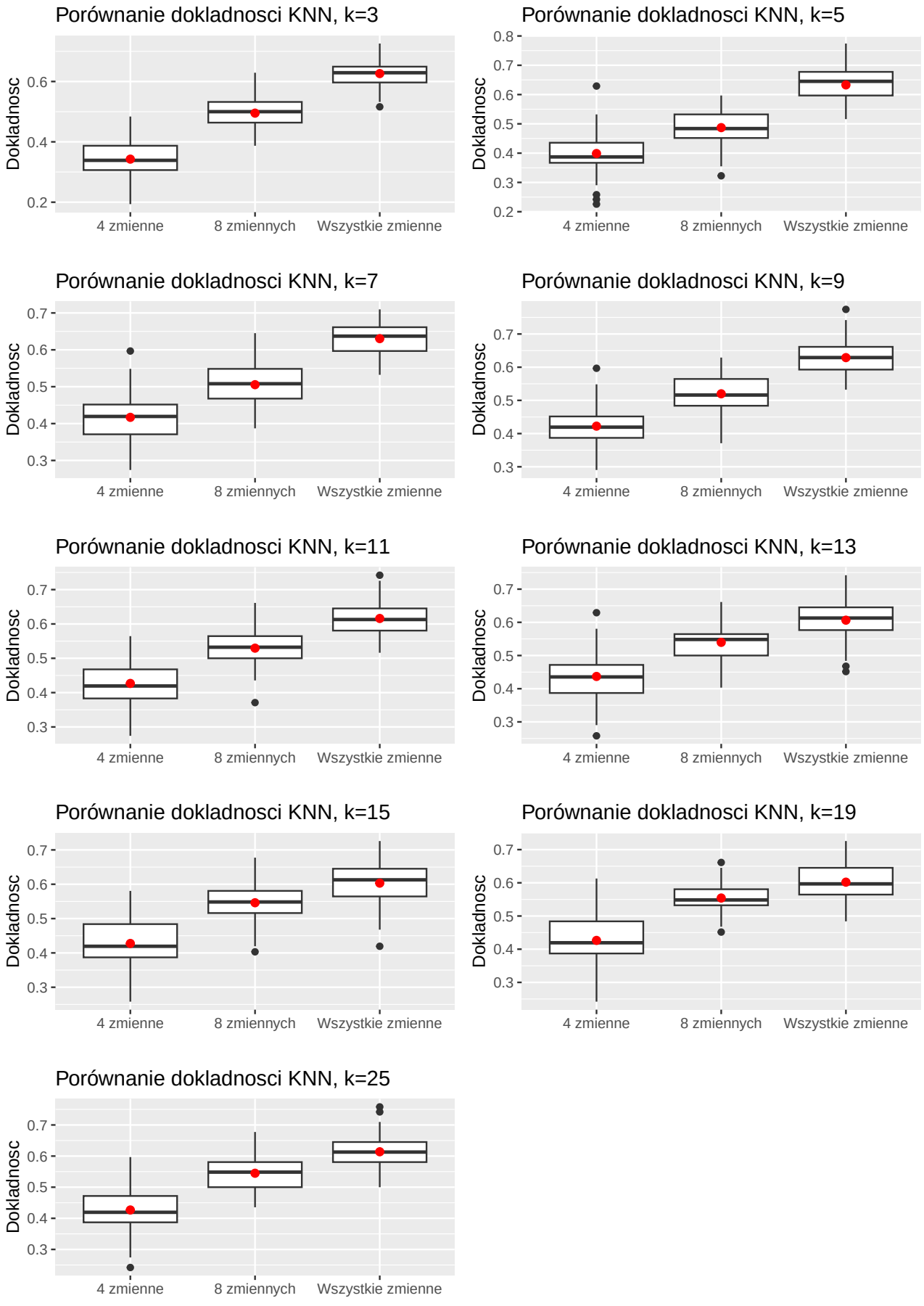
Na rysunku 26 przedstawione zostało porównanie średnich między różnymi wybranymi podzbiorami zmiennych dla każdego z wybranych $k$.




Rysunek 26: Porównanie średnich dokładności dla różnych wyborów cech

Jak możemy zaobserwować powyżej, niezależnie od wybranej wartości $k$, rezultaty zawsze okazywały się jednoznaczne - im więcej zmiennych, tym lepiej dla metody $KNN$ tzn. lepsze mediany dokładności otrzymaliśmy, czyli tak samo jak w przypadku średnich z rysunku 26. Ponadto widzimy, że dla każdego z wybranych podzbiorów zmiennych, wartość optymalna $k$ jest inna. Dla wszystkich zmiennych jest to $k = 5$, dla podzbioru zawierającego 8 zmiennych - $k = 19$, a dla podzbioru z czterema zmiennymi - $k = 13$. Dodatkowo widzimy, że dla dowolnego $k$ dokładność największa jest dla wszystkich zmiennych, a najmniejsza dla podzbioru 4 zmiennych, zmieniają się tylko różnice między posczególnymi wyborami cech.

Na rysunku 27 w formie wykresów pudełkowych zostały przedstawione rezultaty naszych symulacji. Czerwone kropki odpowiadają wartościom odpowiednich średnich.




Rysunek 27: Porównanie dokładności metody KNN dla różnych k i różnych wyborów zmiennych

Jak możemy zaobserwować na rysunkach powyżej, modele KNN mają bardzo zbliżone rozrzuty niezależnie od wybranego $k$ oraz podzbioru cech. Przeważnie są one największe dla podzbioru 4 zmiennych, ale nie są to duże różnice. Na wykresach pudełkowych widzimy też bardzo mało wartości odstających, co może świadczyć o stabilności tej metody.

## 2.2 Lasy losowe

Kolejną metodą klasyfikacji użytą przez nas będą lasy losowe. Jest to technika z zakresu uczenia maszynowego oparta na aplikacji metody `bagging` dla drzew decyzyjnych. Ponadto, przy każdym losowaniu bootstrapowej próby, losowany jest także podzbiór cech, wykorzystywanych przy budowie pojedynczego drzewa. Dla problemów klasyfikacji, przyjęło się, że dobrym wyborem jest losowanie $\sqrt{p}$, gdzie $p$ oznacza liczbę wszystkich zmiennych w zbiorze danych. Dużą zaletą tej metody jest fakt, że potrafi ona uzwględniać zarówno zmienne ilościowe, jak i jakościowe - porządkowe oraz nominalne - i to bez potrzeby ich transformacji na ich numeryczne wersje.

Przez następujący błąd:

```
## [1] "Error: One or more factor levels in the outcome has no data: '-2' "
```

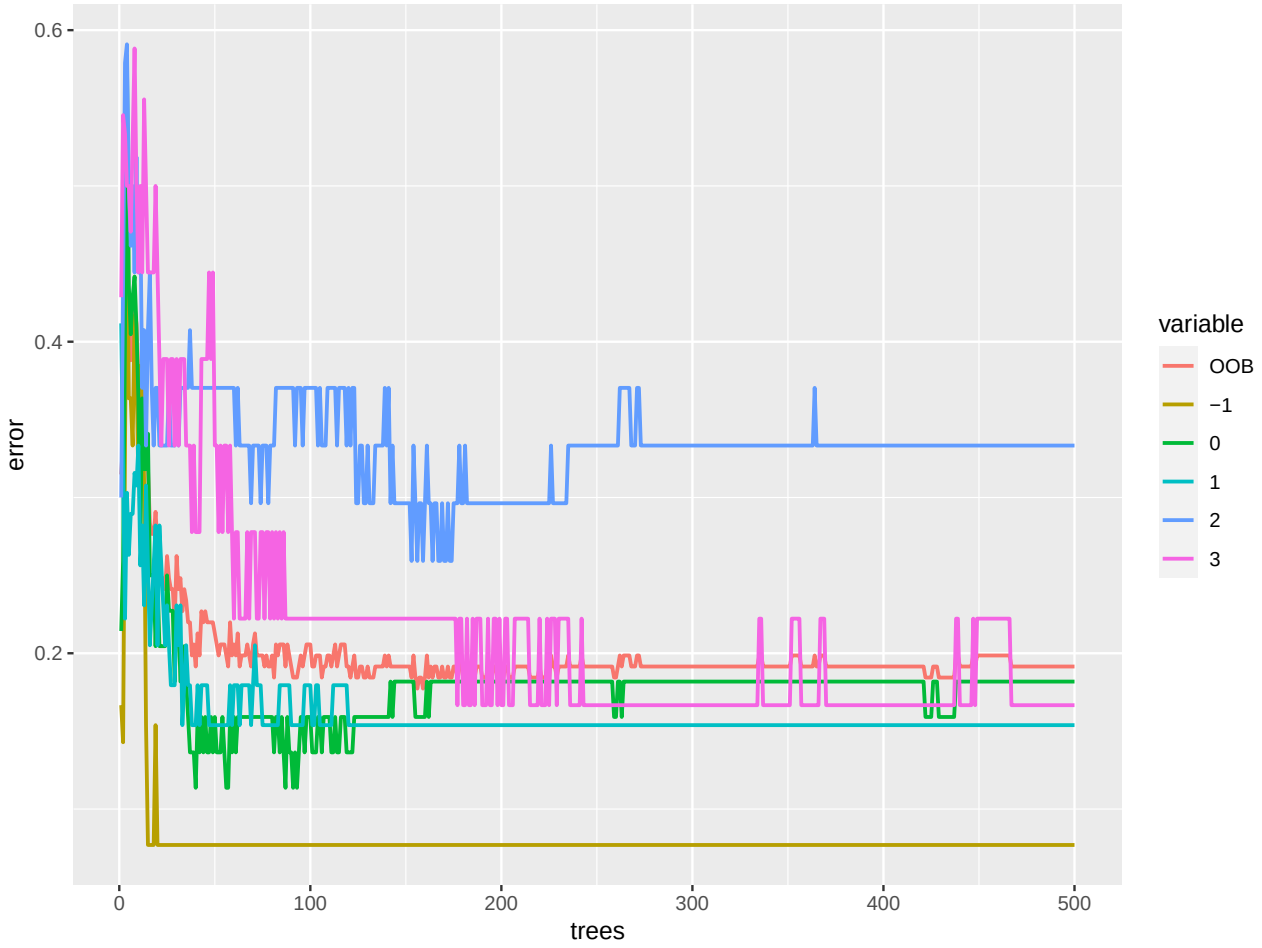
usuwamy z modelu wiersze z wartością zmiennej `symboling` równej $-2$ (oraz usuwamy tenże level z tej zmiennej).

Poniżej przedstawiamy wartości częstotliwości występowania odpowiednich kategorii zmiennej `symboling` po usunięciu z niej wartości $-2$.

```
##
## -1  0  1  2  3
## 22 67 54 32 27
```

Na początku stworzymy jeden model z pomocą lasów losowych. Zbiór treniningowy będzie miał wielkość równą 0.7 rozmiaru naszych danych. Do zbudowania modelu posłużymy się funkcją `trainControl` z pakietu **caret** z 5-krotną walidacją krzyżową oraz funkcją `randomForest` z pakietu **randomForest** ze standardową dla tej funkcji liczbą drzew równej 500.

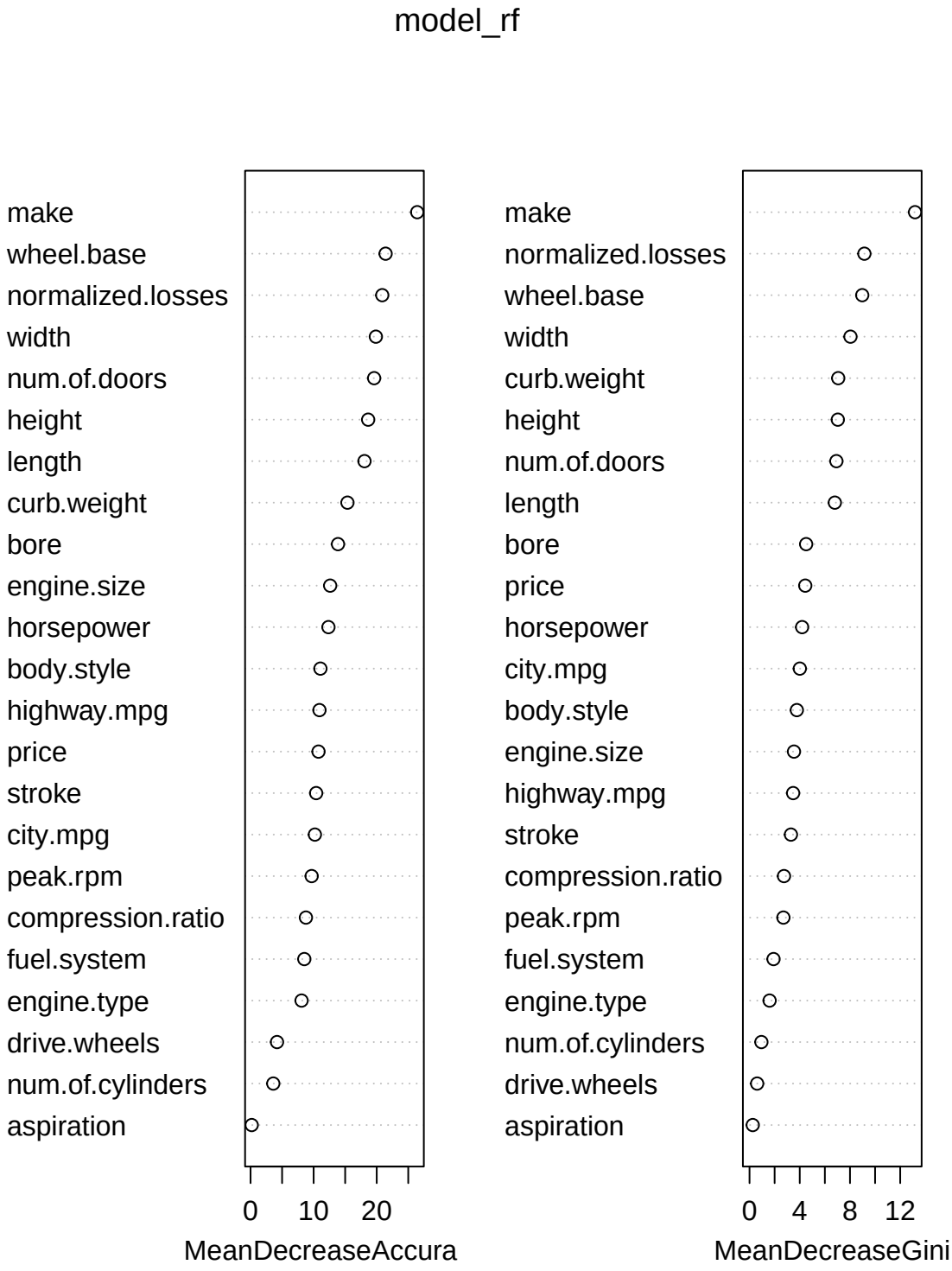
Na rysunku 28 zilustrowane zostały błędy w zależności od ilości drzew lasu losowego dla każdej z kategorii zmiennej `symboling` oraz błąd OOB czyli średni błąd tylko z tych danych, które nie zostały wylosowane do zbioru treningowego.




Rysunek 28: Błędy dla poszczególnych kategorii w zależności od ilości drzew

Na rysunku powyżej widzimy, że od pewnego momentu to jest ok. $ntrees = 120$ wszystkie błędy, a tym samym wartości dokładności stabilizują się i nie zmieniają się wraz z dodaniem dodatkowych drzewek. Dlatego też będzie to maksymalna liczba drzew dla których będziemy budować i porównywać modele. Jednakże widzimy, że już od dużo mniejszej wartości tego parametru, ok. $20 - 30$ wartośći błędów, zmieniają się o względnie niewiele dla większości leveli naszej objaśnianej zmiennej jakościowej.

Dodatkowo, model lasu losowego ma tą zaletę, że możemy przy jego pomocy zrobić ranking wszystkich zmiennych wg. ich istotności dla model (warto zwrócić uwagę na fakt, że ranking będzie obejmował na raz zmienne jakościowe i ilościowe). Na rysunku 29 poniżej przedstawione zostały dwie takie miary. Pierwsza z nich bazuje na sprawdzeniu jak bardzo obniży się dokładność jeśli daną zmienną wykluczymy z modelu, a precyzyjniej mówiąc permutujemy wartości danej zmiennej i sprawdzamy jaki to będzie miało wpływ na skuteczność modelu. Natomiast druga miara określa średni spadek czystości przez podziały danej zmiennej. Jeśli zmienna jest użyteczna, ma tendencję do dzielenia mieszanych węzłów z etykietami na węzły czysto pojedynczej kategorii [10], jednakże jest to dla nas mniej ważna miara.

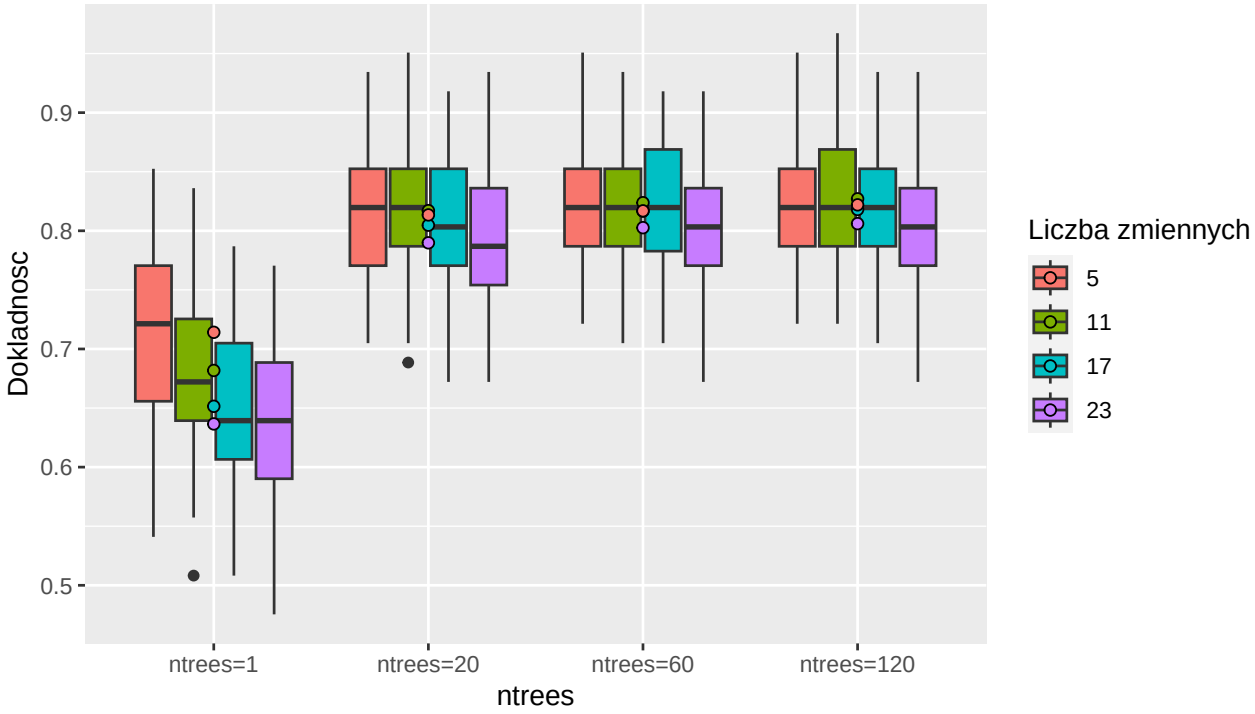



Rysunek 29: Ranking istoności zmiennych

Na rysunku powyżej widzimy ranking istotności wszystkich zmiennych, w szczególności zajmiemy się rankingiem po lewej stronie, opisującym potencjalne spadki dokładności przy usuwaniu zmiennych z modelu. Możemy zaobserwować, że najbardziej istotna jest zmienna `make`, a najmniej `aspiration`. Ponadto, widzimy, że ranking ten potwierdza nasze wnioski z rozdziału 1.5 dotyczące istotności zmienych. Może nie dokładnie odwzorowuje, ale zmienne, które uznaliśmy, że są w grupie najbardziej istotnych, tak samo są i w tej hierarchii. Podobna sytuacja ze

zmiennymi najmniej istotnymi.

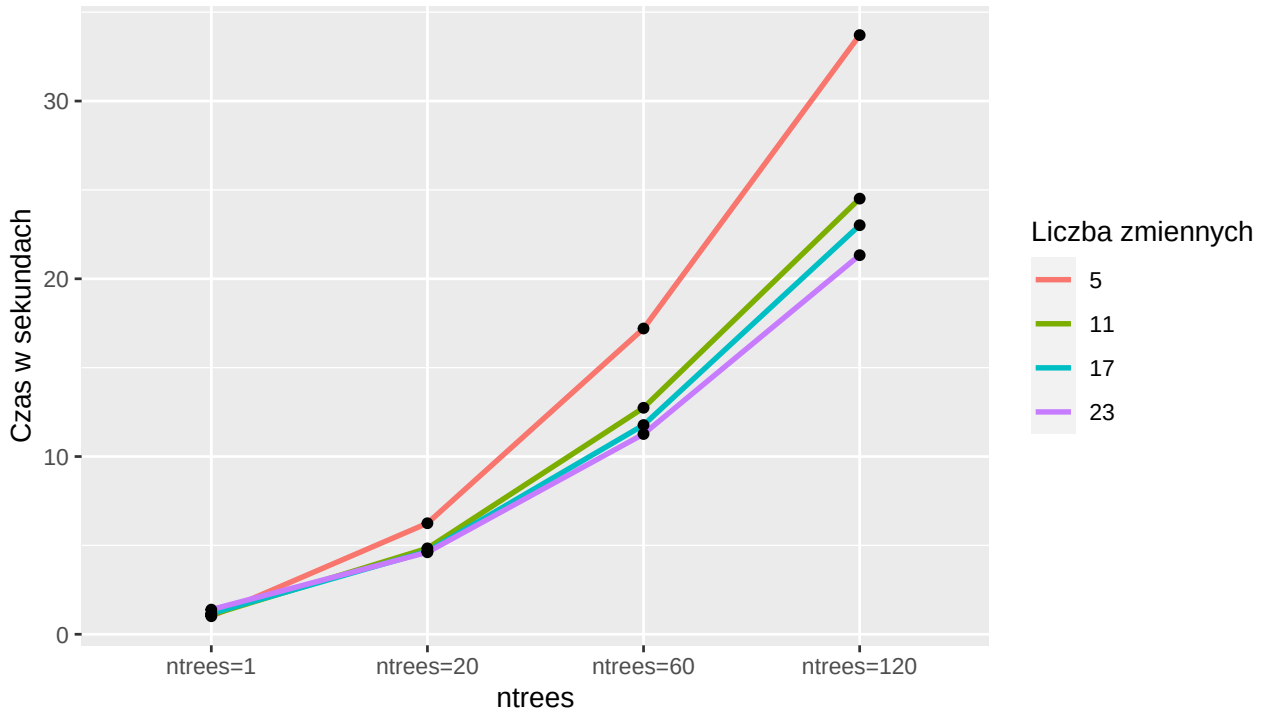
Poniżej na rysunku 30 zostały umieszczone wykresy pudełkowe dla dokładności metody lasów losowych pogrupowane ze wzlędu na liczbę użytych drzew w modelu oraz liczbę zmiennych. Różnobarwne kropki na środku każdej z wartości parametru *ntree* odpowiadają odpowiednim wartościm średnim dla różnych liczb zmiennych.




Rysunek 30: Porównanie dokładności metody lasów losowych dla różnej liczby drzewek i różnych podzbiorów zmiennych

Jak możemy zaobserwować powyżej, jedynie dla wartości $ntrees = 1$, czyli dla pojedynczego drzewa, dokładność jest znacząco gorsza, ale i tu plasuje się na poziomie między 0.6, a 0.7 co sprawia, że nawet ta wersja wdrożenia metody daje lepsze rezultaty niż niektóre metody (np. $KNN$ czy $LDA$, którą zaraz omówię). Dla pozostałych wartości *ntree* wszystkie wartości średnie dokładności oscylują koło 0.8, na oko w przedziale $(0.77, 0.83)$. Jeśli chodzi o wartości średnie dla różnych podzbiorów zmiennych, to możemy stwierdzić, że dla każdej wartości *ntrees* najgorzej wypada metoda uwzględniająca wszystkie zmienne. Dla $ntrees = 1$ zdecydowanie najlepiej działa metoda uwzględniająca najmniej, czyli 5 zmiennych, dla $ntrees = 20$, $ntrees = 60$ i $ntrees = 120$ podzbiory biorące pod uwagę 5, 11 i 17 zmiennych zwracają bardzo zbliżone rezultaty. Jeśli chodzi o $IQR$, to możemy stwierdzić, że są one większe dla podzbiorów uwzględniających 17 i 23 zmienne, niż dla podzbiorów z 5 i 11 zmiennymi, ale różnice nie są szczególnie znaczące.

Poniżej na rysunku 31 przedstawione zostały nieuśrednione czasy obliczeniowe dla wszystkich badanych liczb drzewek oraz podzbiorów cech. Pamiętajmy, że w symulacji obliczenia powtarzaliśmy 100 razy.




Rysunek 31: Porównanie czasów obliczeń lasów losowych dla różnej liczby drzewek i różnych podzbiorów zmiennych dla 100 powtórzen w symulacji

Na wykresie powyżej widzimy, że wszystkie czasy obliczeniowe zwiększają się wraz z dodawaniem kolejnych drzewek, co jest jak najbardziej logicznym rezultatem. Dodatkowo - co ciekawe - obserwujemy, że im zmiennych, tym więcej czasu algorytm potrzebował do obliczeń, co na pierwszy rzut oka wydaje się nieoczywiste.

## 2.3 LDA

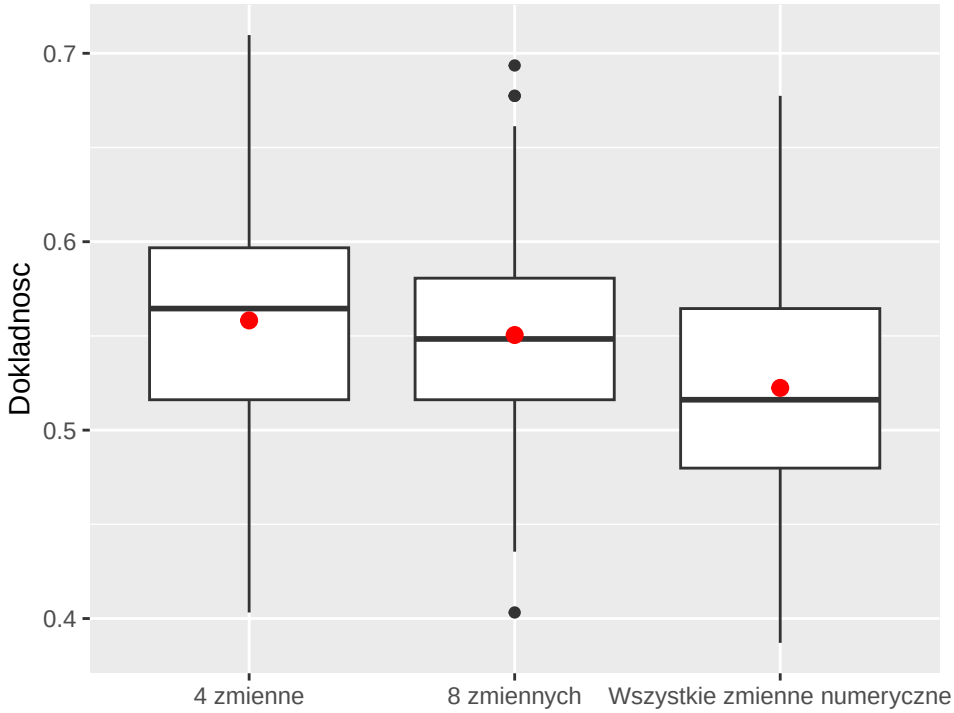
W tym podrozdziale wykorzystana zostanie liniowa analiza dyskryminacyjna. Z racji charakterystyki tego klasyfikatora, do jego konsturkcji posłużymy się tylko i wyłącznie zmiennymi ilościowymi. Próbowałem przetransformować zmienne jakościowe/ich pewne podzbiory na zmienne numeryczne (one hot encoding), jednak tylko pogarszało to otrzymane rezultaty. Prawdopodobnie przez fakt, zwiększenia $p$ czyli wymiaru danych.

Do tworzenia modeli $LDA$ będziemy posługiwać się funkcją `lda` z pakietu **MASS**. Sprawdzimy modele dla:

- Wszystkich zmiennych numerycznych
- 8 zmiennych: `normalized.losses`, `height`, `length`, `wheel.base`, `curb.weight`, `bore`, `city.mpg`, `highway.mpg`
- 4 zmiennych: `normalized.losses`, `height`, `length`, `wheel.base`

Próbki testowe będą standardowo w naszych badaniach miały 0.7 wielkości zbioru danych. Badaną metryką będzie dokładność, a nasze obliczenia wykonamy 100-krotnie, a następnie wyniki uśrednimy.

Na rysunku 32 przedstawione zostały wykresy pudełkowe dla poszczególnych podzbiorów naszych danych. Czerwone punkty odpowiadają odpowiednim wartościom średnim.




Rysunek 32: Porównanie dokładności klasyfikacji LDA

Na podstawie tych wykresów, możemy stwierdzić, że przy ocenie dokładności dla metody *LDA* najlepiej wypadł najmniejszy podzbiór obejmujący tylko 4 zmienne, a najgorzej największy, obejmujący wszystkie zmienne ilościowe. Oprócz tego, widzimy, że rozrzuty dla poszczególnych podzbiorów są porównywalnej wielkości.

## 2.4 QDA

Jeżeli chodzi o metodę QDA, to po pierwszej nieudanej próbie i błędzie dotyczącym jakiejś za małej grupy, usunąłem z danych te wiersze, w których zmienna `symboling` przyjmuje wartość $-2$ (były 3 takie przypadki). Mimo usunięcia tych wierszy oraz wartości $-2$ z poziomów zmiennej jakościowej i tego, że po tej modyfikacji zmienna `symboling` przyjmowała takie wartości z odpowiednią częstością:

```
##
## -1  0  1  2  3
## 22 67 54 32 27
```

funkcja `qda` z pakietu `MASS` dalej zwracała błąd "some group is too small for 'qda' ".

## 2.5 Wielomianowa regresja logistyczna

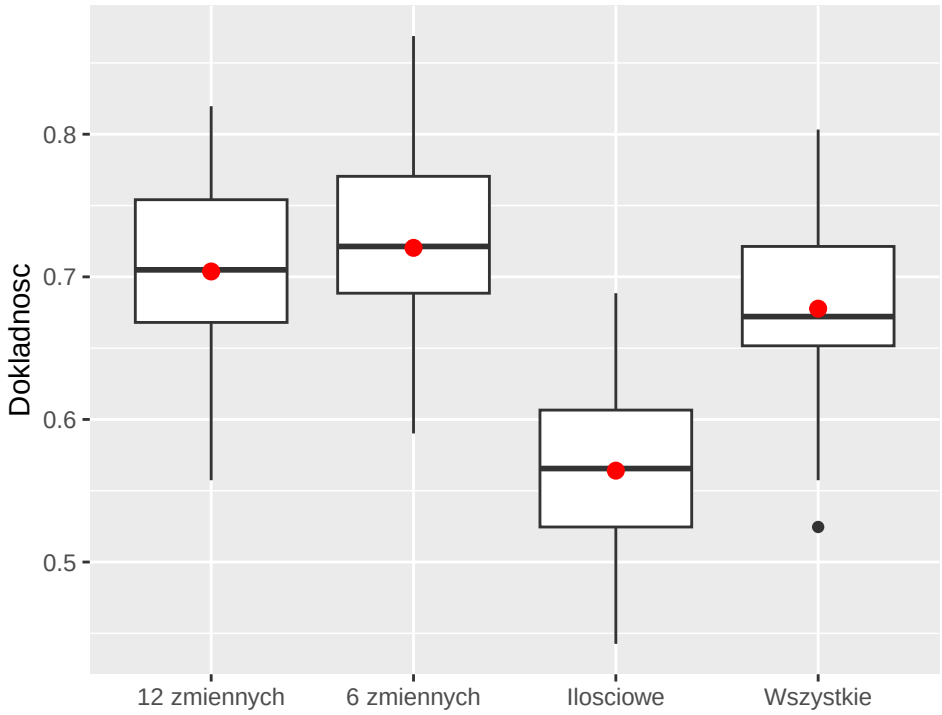
uwaga: ze względu na wyskakujące ostrzeżenia i wartości odstające, metoda klasyfikacji na bazie wielomianowej regresji logistycznej również wykonana jest na danych z usuniętymi wierszami w których zmienna `symboling` przymuje wartość $-2$.

Do tworzenia takich modeli użyjemy funkcji `multinom` z pakietu **nnet**. Zbudujemy modele dla następujących podbziorów:

- Wszystkich zmiennych
- Tylko zmiennych ilościowych
- 12 zmiennych: `make`, `wheel.base`, `num.of.doors`, `normalized.losses`, `height`, `length`, `bore`, `curb.weight`, `horsepower`, `city.mpg`, `highway.mpg`, `price`
- 6 zmiennych: `make`, `wheel.base`, `num.of.doors`, `normalized.losses`, `height`, `length`

Zauważmy, że wybrane podzbiory z 12 i 6 zmiennymi są mieszanką zmiennych ilościowych i jakościowych, oczywiście wybraną mniej więcej wg. hierarchii istotnośmi zmiennych.

Poniżej, na rysunk 33 za pomocą wykresów pudełkowych zostały przedstawione wyniki naszej symulacji z użyciem wielomianowej regresji logistycznej, a odpowiednie czerwone punkty są równe wartościom średnim.




Rysunek 33: Dokładność klasyfikacji regresji logistycznej

Widzimy na rysunku powyżej, że zdecydowanie najgorzej poradził sobie model w ogólnie nie uwzględniający żadnych zmiennych jakościowych. Możemy przypuszczać, że to przez pominięcie bardzo istotnych zmiennych `make`, `num.of.doors`. Poza tym stwierdzamy, że dla "mieszanek" zmiennych jakościowych i ilościowych uwzględnionych w modelu, najlepiej poradził sobie model zawierający najmniej czyli 6 zmiennych, a najgorzej model uwzględniający wszystkie zmienne. Dodatkowo zauważmy, że dla podzbiorów 6 i 12 zmiennych rozrzut jest mniejszy niż w pozostałych dwóch przypadkach.

## 2.6 KDA

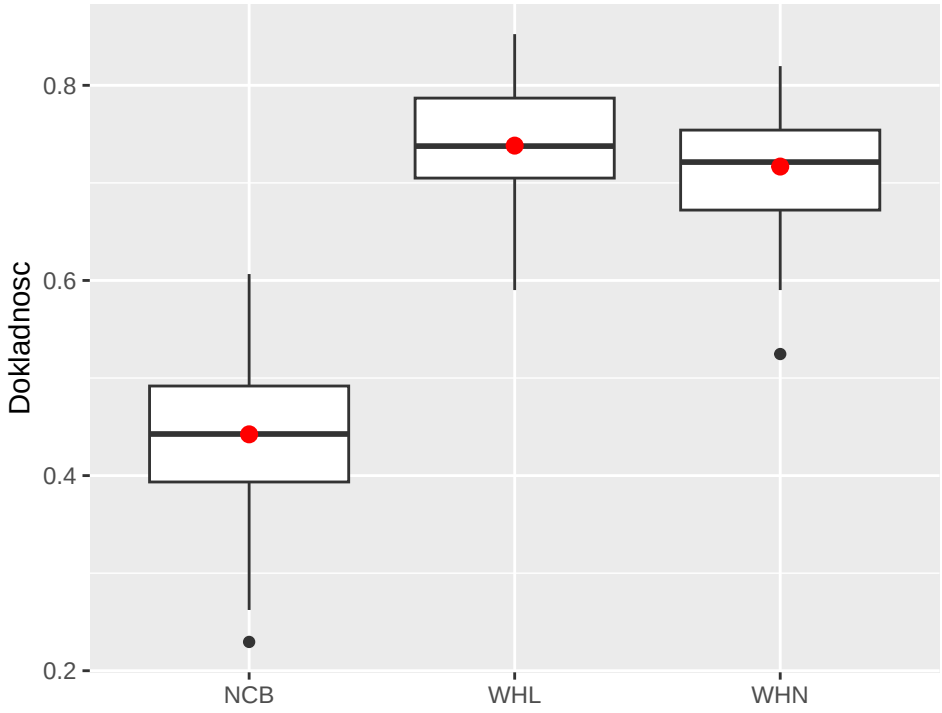
Następną metodą, której użyjemy do klasyfikacji będzie jądrowa analiza dyskryminacyjna czyli $KDA$. Do tworzenia takich modeli użyjemy funkcji `kda` z pakietu **ks**. Warto przypomnieć,

że metody $KDA$ nie używa się w praktyce dla $p > 6$, dlatego my też mocno okroimy nasze dane do $p = 3$ oraz że będziemy używać tylko zmiennych ilościowych. Wybraliśmy 3 następujące podzbiory zmiennych:

- `wheel.base`, `height`, `length` ($WHL$)
- `wheel.base`, `height`, `normalized.losses` ($WHL$)
- `normalized.losses`, `curb.weight`, `bore` ($NCB$)

Tak jak wcześniej, będziemy badać dokładność, a losowanie zbioru treningowe oraz obliczenia przeprowadzimy 100 razy.

Na rysunku 34 przedstawione zostały wykresy pudełkowe dla naszych różnych wyborów cech. Czerwone kropki odpowiadają wartościom średnim.




Rysunek 34: Porównanie dokładności klasyfikacji na podstawie KDA

Na rysunku powyżej widzimy, że najgorzej poradził sobie podzbiór ($NCB$), i jeśli chodzi o średnią dokładność, i jeśli chodzi o stabilność, co jest dość logiczne, ponieważ wybrane tam zmienne w naszej analizie nie były w czołówce najbardziej istotnych. Dwa pozostałe zbiory poradziły sobie dużo lepiej, w szczególności ($WHL$), który ma najlepszą dokładność oraz najmniejszy rozrzut otrzymanych wyników.

## 2.7 Naive Bayes

Ostatnią metodą zaimplementowaną dla naszych danych będzie naiwny klasyfikator bayesowski, który zakłada, że nasze zmienne objaśniające są niezależne. Oczywiście jest to faktycznie bardzo "naiwne" założenie, w rozdziale 1.5.5 pokazaliśmy, że nasze zmienne są zależne.

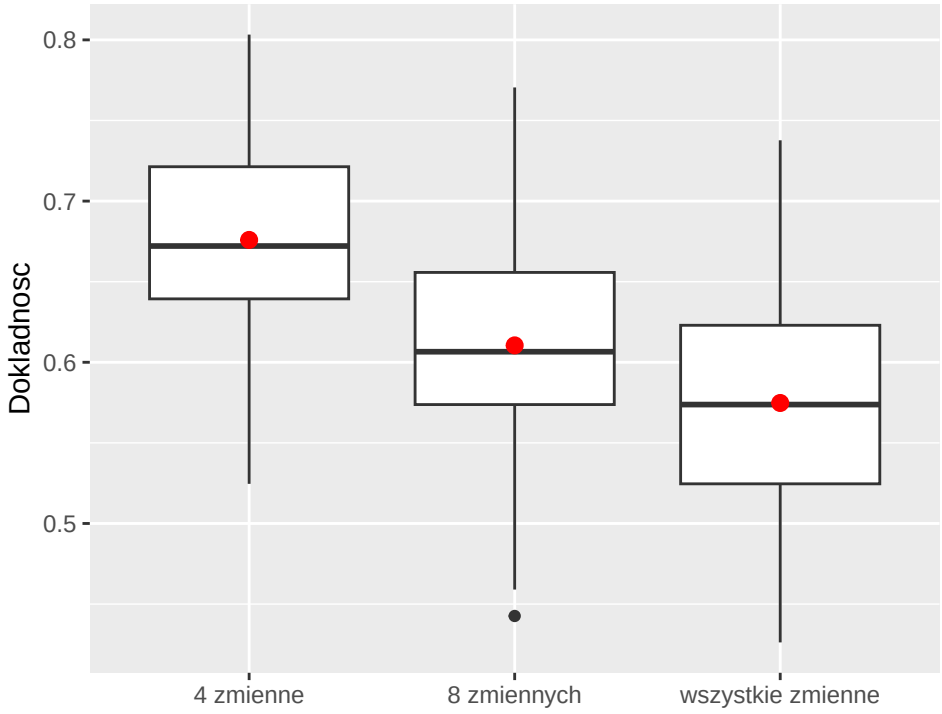
W tym przypadku również użyjemy danych z pominięciem `symboling` $= -2$. Uwzględnimy tylko zmienne numeryczne.

Do tworzenia modeli naiwnego klasyfikatora bayesowskiego będziemy posługiwać się funkcją `NaiveBayes` z pakietu **klaR**. Sprawdzimy modele dla:

- Wszystkich zmiennych numerycznych
- 8 zmiennych: `normalized.losses`, `height`, `length`, `wheel.base`, `curb.weight`, `bore`, `city.mpg`, `highway.mpg`
- 4 zmiennych: `normalized.losses`, `height`, `length`, `wheel.base`

Tak jak poprzednio, będziemy mierzyć dokładność, a symulacje przeprowadzimy 100-krotnie dla każdego podzbioru.

Poniżej na rysunku 35 na wykresach pudełkowych przedstawione zostały wyniki uzyskyskane metodą naiwnego klasyfikatora bayesowskiego, a czerwone punkty sa równe odpowiadającym wartościom średnim.




Rysunek 35: Porównanie dokładności metodą Naive Bayes

Na rysunkach powyżej widzimy, że dokładność naiwnego klasyfikatora bayesowskiego jest tym większa, im mniejsza liczba wybranych cech, a jeśli chodzi o stabilność, to rozrzuty w poszczególnych grupach są porównywalne.

# 3 Podsumowanie I części projektu

Podsumowując, na samym początku, przy wdrażaniu metod klasyfikacyjnych do konkretnego zagadnienia trzeba pamiętać o ich założeniach, o tym, że niektóre metody takie jak $KNN$

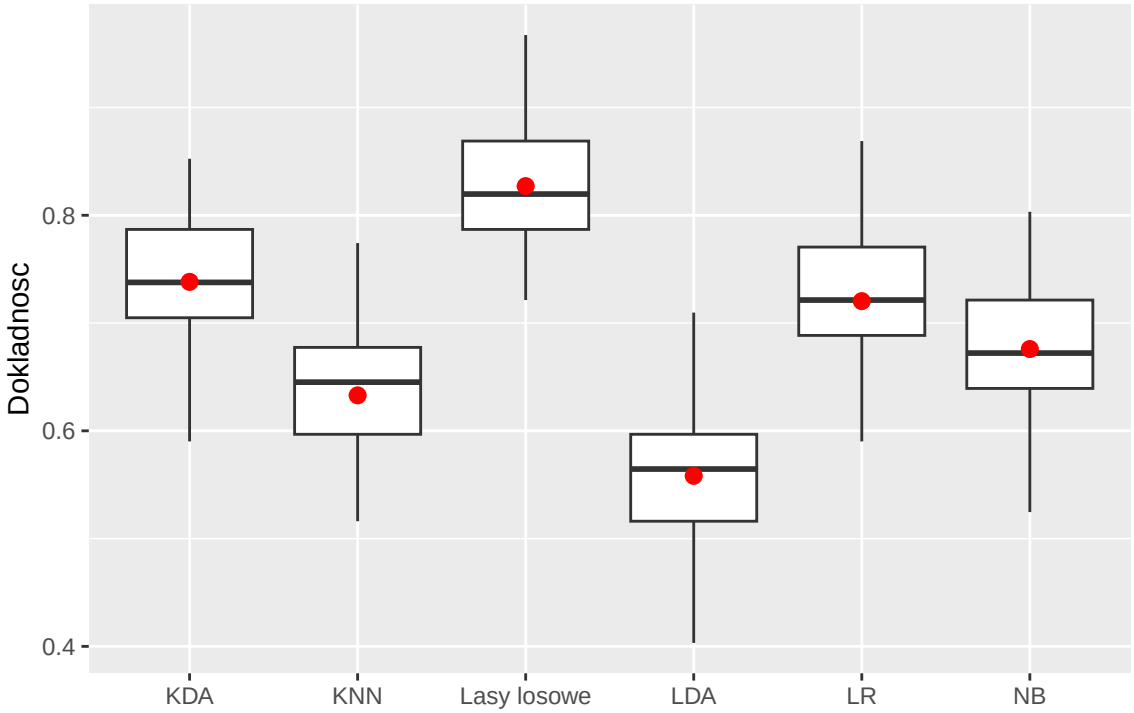
czy $LDA$ obsługują zmienne numeryczne lub o tym że różne metody jak np. $QDA$ czy lasy losowe moga mieć problem jeśli jedna z kategorii naszej zmiennej objaśnianej występuje w niezwykle małej liczbie. Dodatkowo metody mogą mieć inne różne uwarunkowania np. metoda jądrowej analizy dyskryminacyjne w praktyce może być użyta tylko dla małych wymiarów, dla powiedzmy $p < 7$.

Po drugie, przy implementacji każdej metody trzeba pamiętać o tym, że dla każdej metody trzeba rozważyć różne parametry (np. $k$ w metodzie k-najbliższych sąsiadów) lub różne podzbiory zmiennych, ponieważ wyniki dla różnych wersji wdrożenia każdej metody mogą być znacząco różne. Dodatkowo, oczywiście należy pamiętać, że dla różnych zbiorów danych o różnej charakterystyce, wyniki tych samych metod moga się różnić.

Po trzecie, trzeba pamiętać o różnorakich miarach oceny naszych algorytmów. W naszej ocenie kierowaliśmy się tylko oceną dokładności, ale w różnych sytuacjach wybraną miarą porównawczą może być np. czułość czy swoistość. Dodatkowo trzeba pamiętać o stabilności otrzymanego rozwiązania - która jest coraz ważniejszym czynnikiem w ocenie metod z zakresu data mining - czyli sprawdzeniu tego jak różne wyniki daje nasza metoda przy niewielkiej zmianie danych.

Na rysunku 36 zilustrowane zostało porównanie wykresów pudełkowych dla wszystkich metod. Analizując średnie dokładności oraz stabilności metod, z każdej z nich wybraliśmy jej najlepszą wersję implementacji tj:

- KNN - Wszystkie zmienne, $k = 5$
- Lasy losowe - 120 drzewek, 11 zmiennych
- LDA - 4 zmienne
- LR - 6 zmiennych
- KDA - zmienne $WHL$
- NB 4 zmienne




Rysunek 36: Porównanie dokładności różnych metod

Powyżej widzimy, że w naszej analizie porównawczej kilku metod, wyniki jeśli chodzi o średnie i mediany dokładności są bardzo podobne. Spośród wybranych algorytmów wyraźnie najlepiej poradziły sobie lasy losowe, które jako jedyne uzyskały ponad wartości większe niż 0.8 jeśli chodzi o średnie i mediany. Kolejnymi najlepszymi metodami były $KDA$ oraz wielomianowa regresja logistyczna, które moga się poszczycić medianami i średnimi większymi od 0.7, przy czym $KDA$ obie te wartości ma większe. Kolejne metody - $KNN$ oraz naiwny klasyfikator bayesowski mają średnie oraz mediany ponad 0.6, jednakże naiwny klasyfikator bayesowski obie te metryki miał na lepszym poziomie. Najgorzej w naszym zestawieniu wypadła metoda $LDA$, której nawet najlepsza wersja implementacji nie zdołała uzysać rezultatów na poziomie 0.6 (dokładniej średnia oraz mediana z zakresu $(0.55, 0.6)$, jednakże trzeba pamiętać, że to i tak dość dobry rezultat przy zagadnieniu klasyfikacji dla 5 kategorii (w przypadku $LDA$, w przypadku niektórych innych metod - 6 klas), gdzie dokładność przy losowaniu kategorii zmiennej objaśnianej powinno dawać rezultaty w przybliżeniu 0.16.

# 4 Analiza skupień, klasteryzacja

## 4.1 Wstęp

Analiza skupień jest ważnym zagadnieniem w obszarze data mining. Jest narzędziem służącym do eksploaracji danych, jej celem jest ułożenie obiektów w klastry w taki sposób, aby stopień powiązania obiektów z obiektami należącymi do tego samego klastra był jak największy, a z obiektami z innych klastrów jak najmniejszy. Dodatkowo warto wspomnieć, że analiza skupień często jedynie wykrywa struktury w danych, ale nie wyjaśnia dlaczego takie struktury występują.

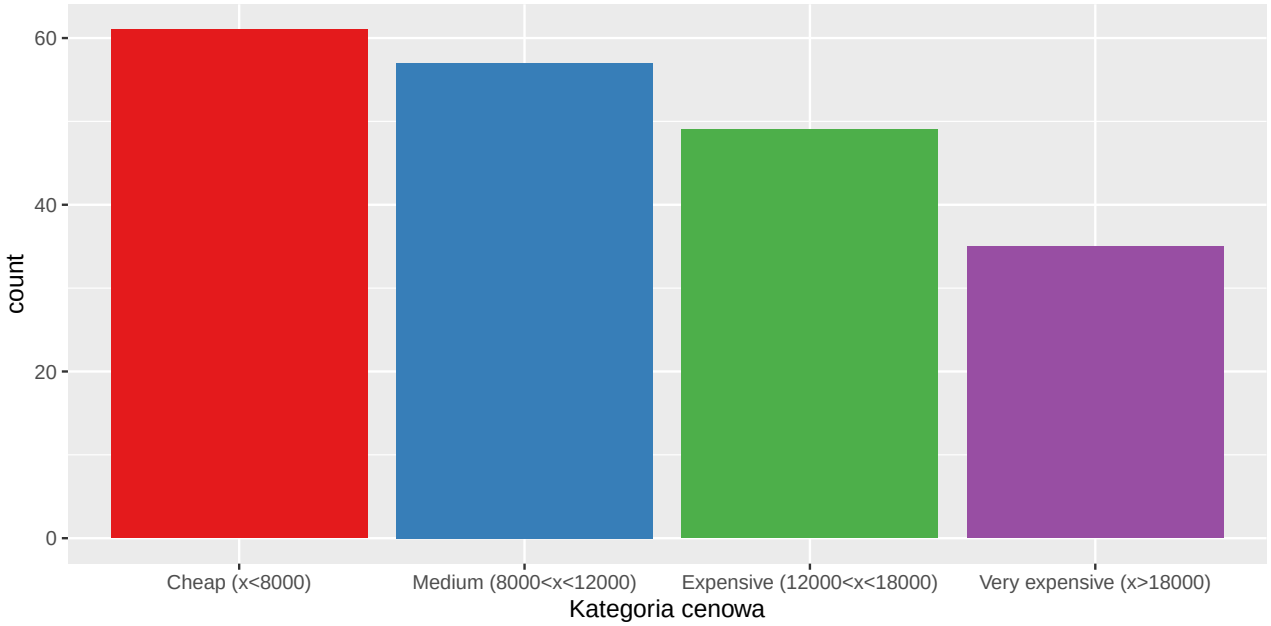
Jeśli chodzi o nasz zbiór danych, na start usuwamy z niego wiersze, gdzie wartości zmiennej `symboling` są równe $-2$, robimy tak z powodu zbyt małej liczności takich obserwacji - jest ich zaledwie 3. (Dodatkowo usuwamy tą wartość z leveli zmiennej typu factor). W analizie skupień nie używa się kategorii zmiennej objaśnianej, dlatego tworzymy dwa podzbiory danych, jeden zawierający wszystkie zmienne poza zmienną `symboling`, a drugi zawierający wszystkie zmienne ilościowe (tym samym również bez zmiennej objaśnianej `symboling`). Zbiór danych ze zmiennymi numerycznymi standaryzujemy. Poniżej fragment kodu przedstawiający te transformacje.

```
data_clust <- data2[!(data$symboling==-2),]
data_clust$symboling <- droplevels(data_clust$symboling)
data_clust_features <- data_clust[, 2:26]
data_num_clust <- data_clust[c("symboling", int_kol, num_kol)]
data_num_clust_features <- data_num_clust[, 2:16]
data_num_clust_features <- as.data.frame( scale(data_num_clust_features) )
data_clust_symboling_real <- data_clust[,1]
data_num_clust_features_UNSCALED <- data_num_clust[, 2:16]
```

Ponadto, korzystając z faktu, że zmienna `symboling` jest zmienną porządkową, tworzymy nowy wektor gdzie grupujemy jej wartości: w jednej grupie znajdują się wartości mniejsze, czyli $-1$ i 0, a w drugiej większe wartości: 1, 2 i 3. Taka konstrukcja przyda nam się w późniejszej części analizy. Poniżej zamieszczony został fragment kodu przedstawiający stworzenie tego wektora.

```
symboling_custom<- as.numeric(data_clust$symboling)
symboling_custom[which(symboling_custom %in% c(1,2))] <- "-1 or 0"
symboling_custom[which(symboling_custom %in% c(3,4,5))] <- "1 , 2 or 3"
```

Tworzymy jeszcze jedną zmiennej na bazie istniejącej. Mianowicie przeprowadzamy dyskretyzacje zmiennej ciągłej `price` w następujący sposób: tworzymy 4 przedziały, które są oddzielane przez liczby 8000, 12000 i 18000. Taka konstrukcja również nam się przyda w późniejszej fazie. Poniżej na rysunku 37 przedstawiony został wykres słupkowy owej zmiennej po takiej transformacji.



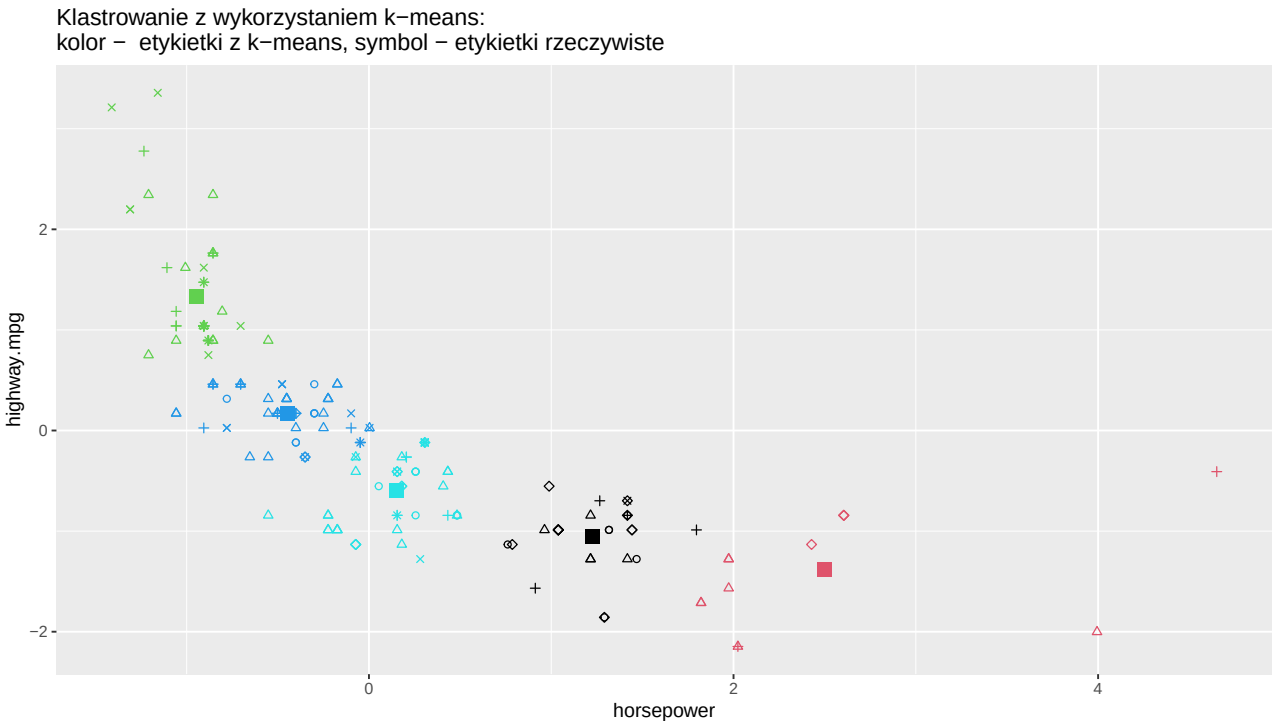

Rysunek 37: Częstotliwość występowania poszczególnych grup zmiennej `price`

## 4.2 Metody grupujace

### 4.2.1 Metoda k -średnich

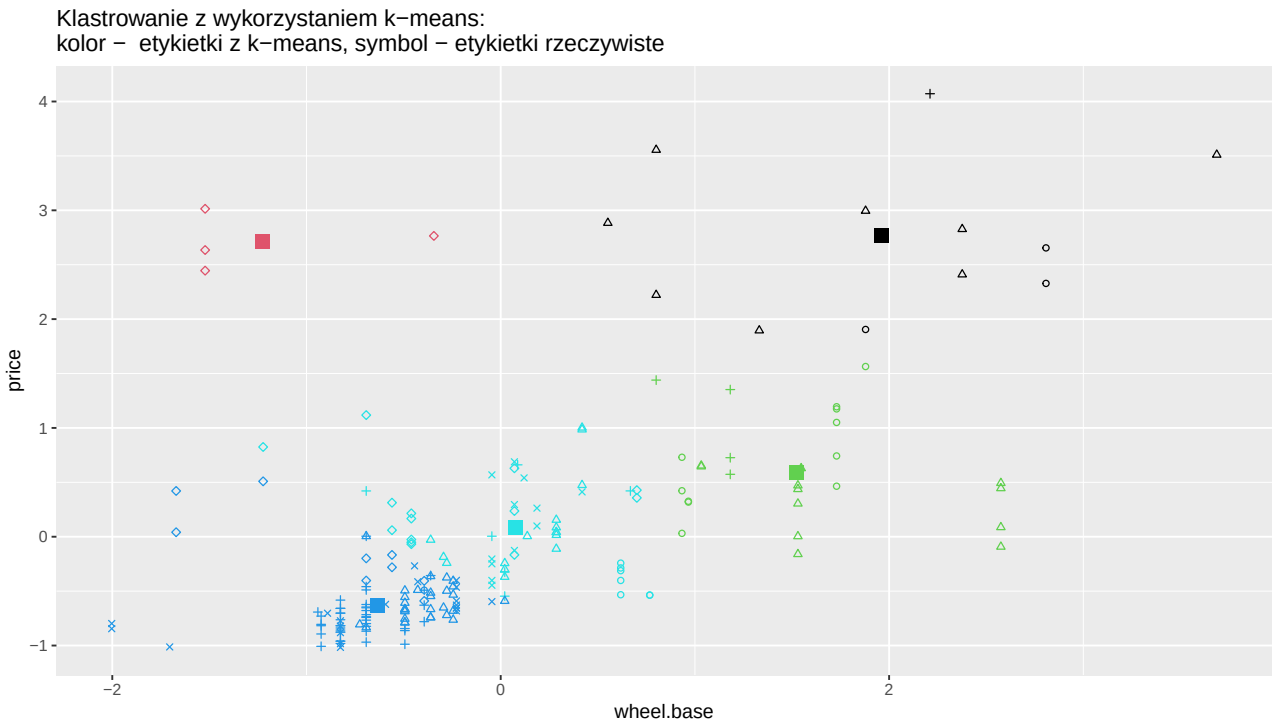
Pierwszą metodą grupującą jakiej użyjemy będzie metoda $k$-średnich ($k - means$). Jest to metoda której na wejściu dostarczamy dane liczbowe (czyli można uwzględnić tylko zmienne numeryczne!), a także $K$ - liczbę klastrów na jaką chcemy podzielić zbiór danych. W inicjalizacji wybieramy $K$ wartości początkowych będących średnimi (centrami skupień). Następnie przyporządkowujemy każdy obiekt do najbliższej mu średniej. Z tak powstałego przyporządkowania wyznaczamy nowe centra skupień (średnie wektorowe). Kroki przyporządkowania i liczenia nowych średnich powtarzamy do spełnienia warunku zbieżności.

Do implementacji metody $k$-means w **R** posłuży nam funkcja `kmeans` [11] z pakietu **stats**. Na rysunku 38 przedstawiona została wizualizacja zastosowania metody $k - means$ na przykładzie zmiennych `horsepower` i `highway.mpg` (skrótowo: $HH$) z rozdzieleniem na $K = 5$ klastrów z maksymalną liczbą iteracji równą 15. Liczba klastrów podyktowana jest oczywiście liczbą kategorii objaśnianej zmiennej `symboling` (po usunięciu z niej kategorii wartości $-2$). Na rysunku zaznaczone sa także większymi, wypełnionymi kwadratami końcowe centra skupień. Ponadto, każda z obserwacji oprócz koloru, ma także przypisany kształt odpowiadający rzeczywistej wartości zmiennej `symboling`.




Rysunek 38: Metoda k-means na dwóch zmiennych: horsepower i highway.mpg oraz z podziałem na 5 klastrów

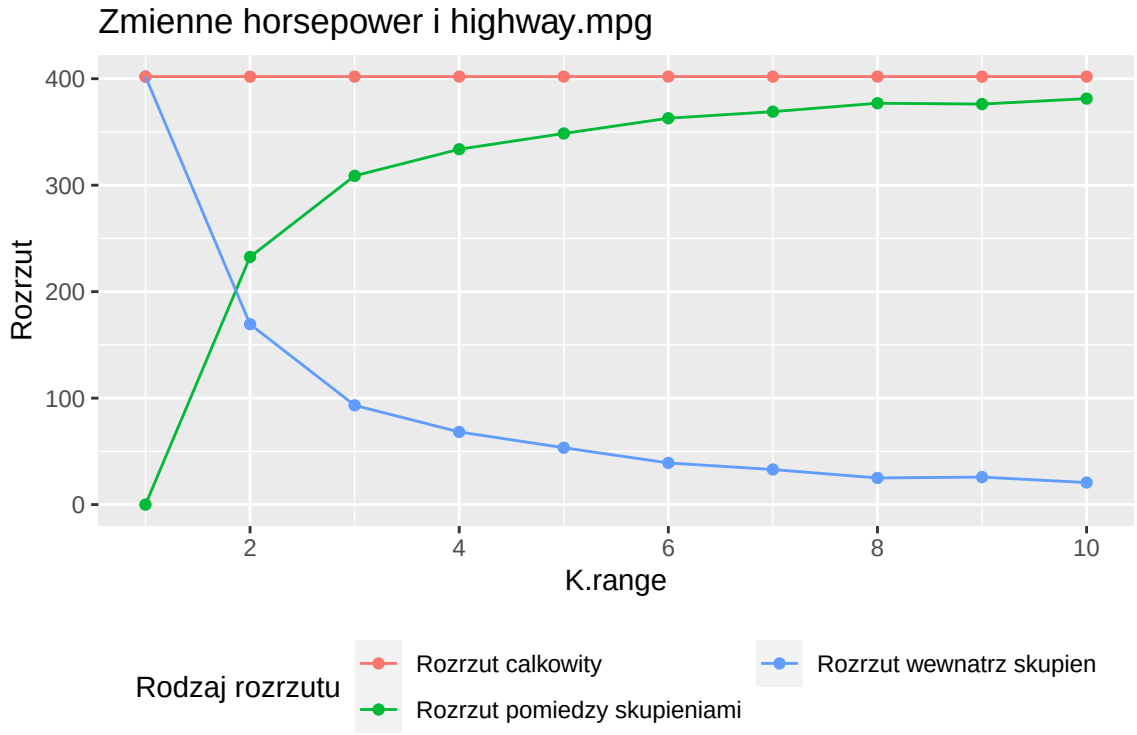
Na rysunku 39 umieszczony został analogiczny wykres, tylko że dla zmiennych `wheel.base` i `price` (skrótowo: $WP$).



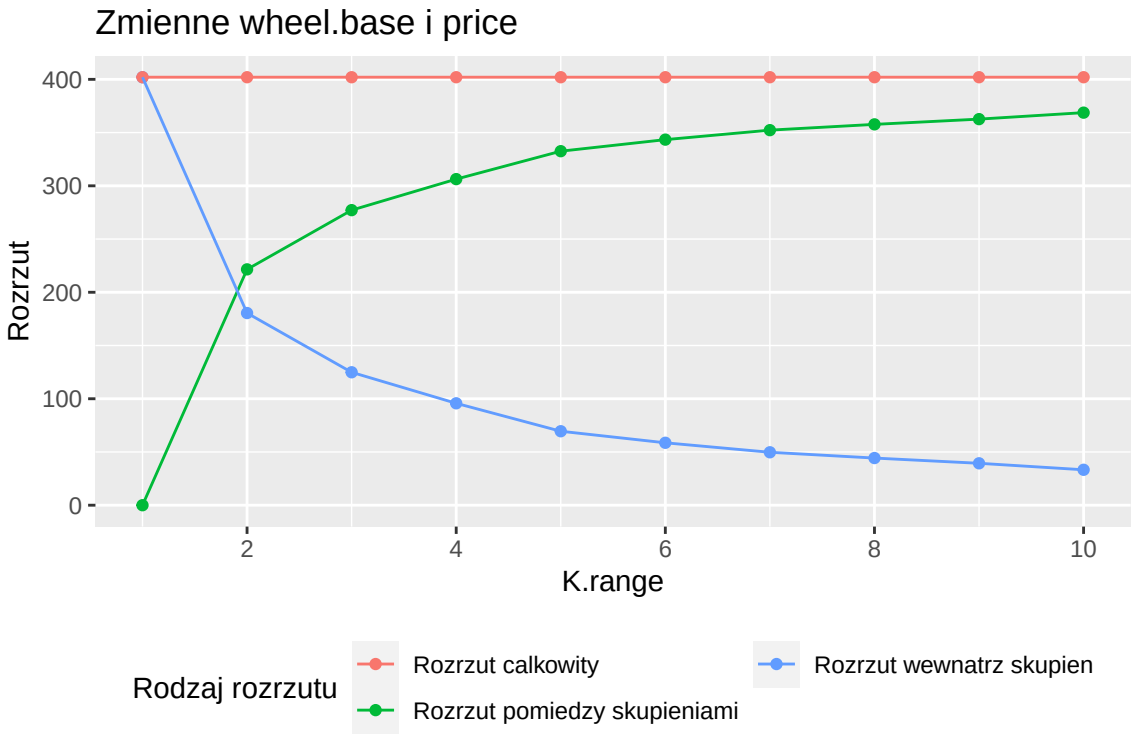

Rysunek 39: Metoda k-means na dwóch zmiennych: wheel.base i price oraz z podziałem na 5 klastrów

Na obu rysunkach powyżej widzimy jak algorytm $k - means$ stworzył podziały na 5 grup. Widzimy, że te podziały wyglądaja na sensowne tzn. obszary odpowiadające wszystkim klastrom są rozłączne. W przypadku zmiennych $HH$ klastry są bardziej równoliczne. Co ciekawe, dla tej pary zmiennych widzimy, że obserwacje rozproszone są na wykresie punktowym bardziej równomiernie, niż dla zmiennych $WP$, gdzie punkty są bardziej skoncentrowane w okolicach `wheel.base`$= -0.3$, `price`$= -0.7$ niż w obszarach bardziej odległych od tego miejsca. Dlatego też skupienia dla tych zmiennych można ocenić jako mniej zwarte, w przypadku zmiennych $HH$ skupienia są dużo bardziej zwarte dla wszystkich grup, z wyjątkiem 2 punktów odstających w klastrze o kolorze czerwonym.

Żeby formalniej ocenić lub porównać jakość klasteryzacji, możemy użyć w tym celu różnych wskaźników. Przykładami takich miar są rozrzut wewnątrz skupień (ang. within-cluster scatter) czy rozrzut pomiędzy skupieniami (ang. between-cluster scatter). Tak jak wskazują nazwy, pierwsza z tych miar wskazuje jak duży jest rozrzut między obiektami z tych samych klastrów, a druga ocenia rozrzut między obiektami z różnych klastrów. Poniżej na rysunkach 40, 41 przedstawione zostały porównania obu tych miar oraz totalnego rozrzutu (który jest sumą tych dwóch miar) dla różnych wartości $k$- klastrów w zakresie od 1 do 10 dla zmiennych $HH$ oraz $WP$.



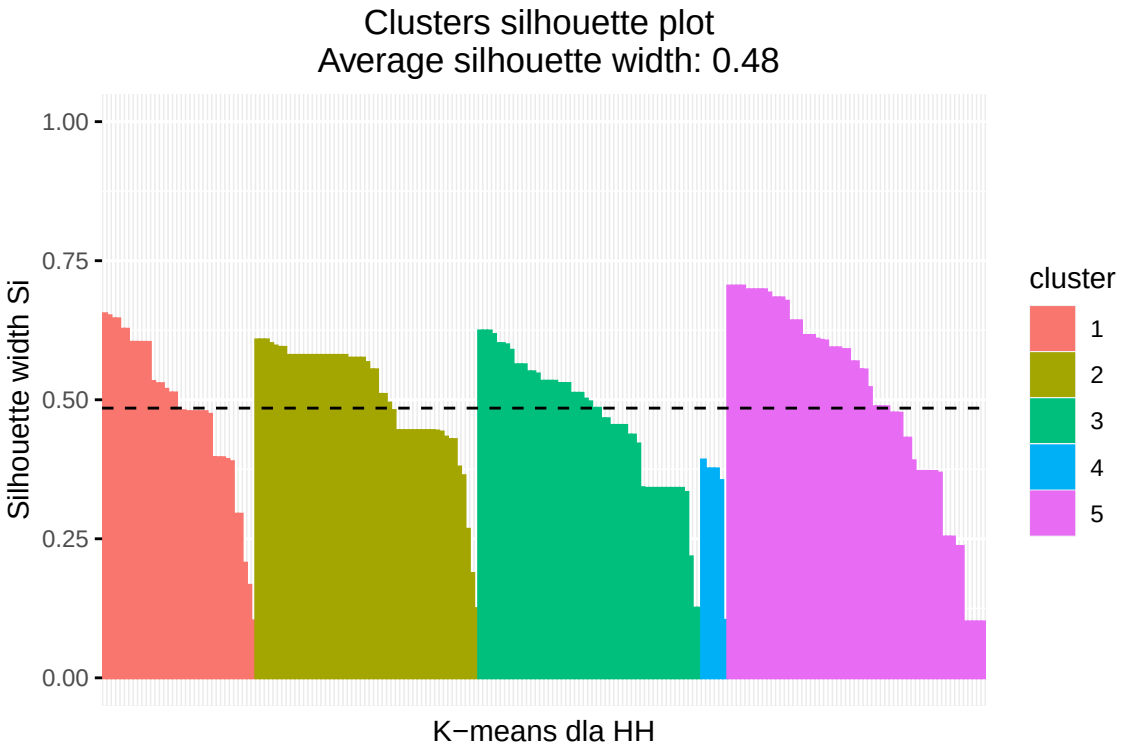

Rysunek 40: Porównanie różnych rozrzutów dla różnch liczb klastrów




Rysunek 41: Porównanie różnych rozrzutów dla różnch liczb klastrów

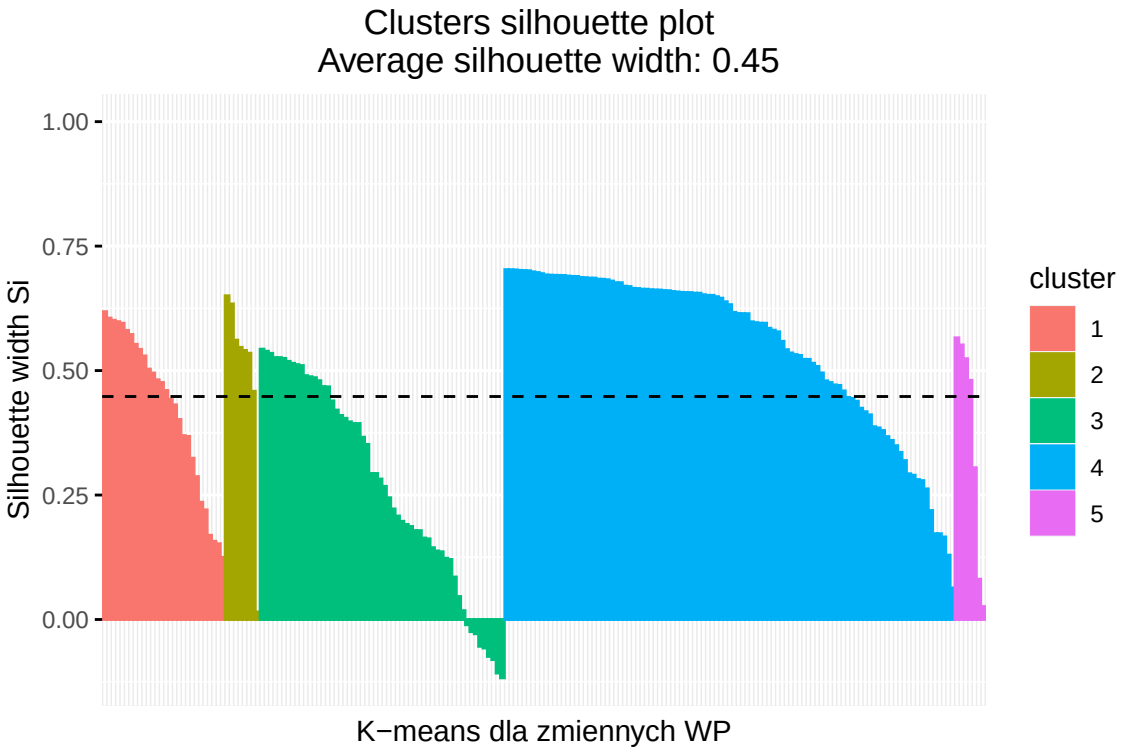
Możemy zauważyć, że oba wykresy są do siebie bardzo podobne. Co oczywiste wraz ze wzrostem parametru $k$, maleje rozrzut wewnątrz skupień - są bardziej zwarte, a rośnie rzorzut między skupieniami. Widzimy, że dla $k = 5$, dla którego zwizualizowane zostały wyżej oba rozrzuty, dla zmiennych $HH$ rozrzut wewnątrz skupień jest równy ok. 50, a dla zmiennych $WP$ ta wartość jest równa ok. 70, co potwierdza nasze wnioski z analizy wizualnej, że skupienia w przypadku $HH$ są bardziej zwarte.

Innym wskaźnikiem służącym do oceny klasteryzacji jest indeks Silhoutte'a [12]. Wskaźnik ten ocenia każdy obiekt w skali $[-1, 1]$, przy czym wartości im bliżej 1, tym większe prawdopodobieństwo, że obiekt został poprawnie przyporządkowany, a im bliżej $-1$, tym większa szansa, że został on przyporządkowany błędnie. Wartości w okolicach 0 świadczą o tym, że obiekt leży pomiędzy klastrami $A$ i $B$. Poniżej na rysunkach 42 i 43 widzimy wykresy Silhouette dla metody $k$-means dla odpowiednio zmiennych `horsepower` i `highway.mpg` oraz `wheel.base` i `price`. Przed każdym wykresem zamieszczony został fragment kodu z wypisanymi indeksami Silhouette'a dla poszczególnych klastrów grupowania.




Rysunek 42: Wykres Silhouette zmiennych HH i metody k-means

```
## [1] "Średnie wartości indeksów Silhouette dla każdego z klastrów:"
## [1] 0.482 0.506 0.465 0.330 0.502
## [1] "Średnia wartość indeksu Silhouette ogółem: 0.485"
```




Rysunek 43: Wykres Silhouette dla zmiennych WP i metody k-means

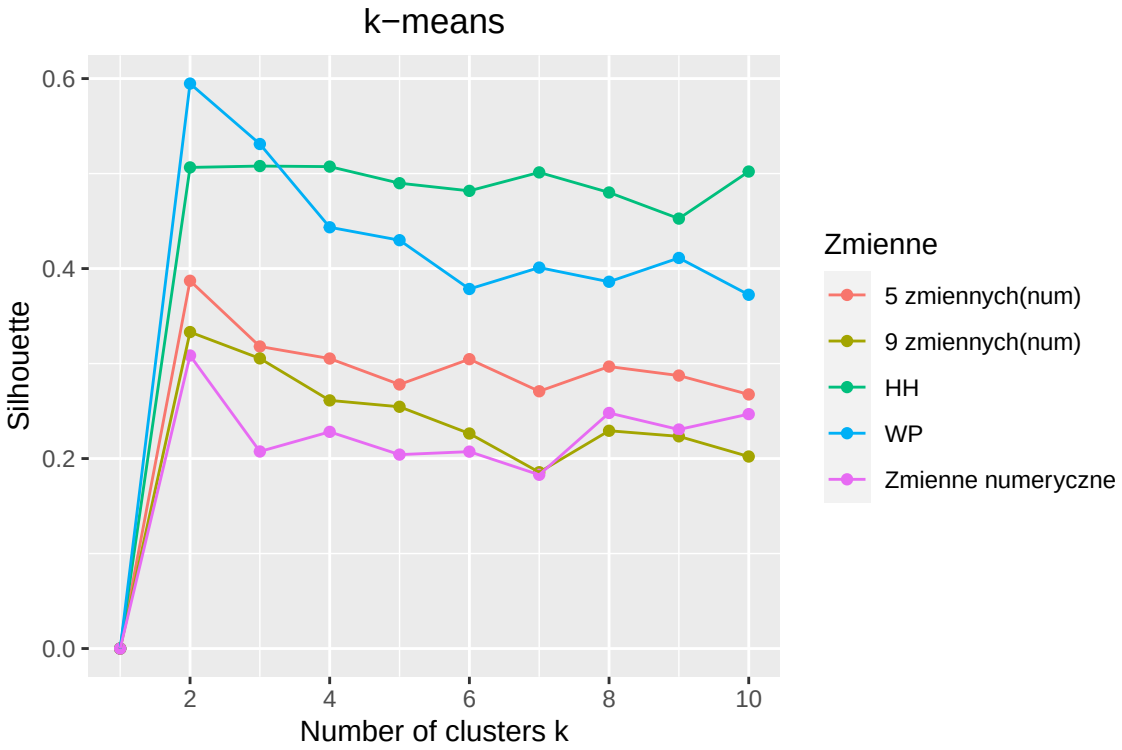
```
## [1] "Średnie wartości indeksów Silhouette dla każdego z klastrów:"
## [1] 0.425 0.493 0.268 0.554 0.362
## [1] "Średnia wartość indeksu Silhouette ogółem: 0.448"
```

Z rysunków oraz przedstawionych wartości liczbowych średnich indeksów dla poszczególnych klastrów powyżej, możemy wyciągnąć kilka wniosków. Po pierwsze i najważniejsze, widzimy, że średnia wartość indeksu Silhouette'a dla zmiennych $HH$ jest większa niż dla zmiennych $WP$. Co więcej, widzimy także, że wartości indeksów dla poszczególnych klastrów dla $HH$ mają mniejszy rozrzut np. średnia wartość w tylko jednym klastrze jest mniejsza niż 0.45. Co więcej, potwierdzają się nasze obserwacje dotyczące tego, że klastry dla przypadku $HH$ są bardziej "równoliczne", mimo jednej mniejszej grupy. Ponadto widzimy, że w przeciwieństwie do zmiennych $WP$, dla $HH$ nie istnieją żadne obiekty, dla których indeks Silhouette'a byłby ujemny.

Teraz porównamy średnie indeksy Silhouette'a dla różnych liczb klastrów $k$ oraz dla następujących podzbiorów zmiennych ilościowych:

- zmienne `horsepower` i `highway.mpg` ($HH$)
- zmienne `wheel.base` i `price` ($WP$)
- zmienne `wheel.base`, `height`, `length`, `normalized.losses`, `curb.weight` (5 zmiennych)
- zmienne `wheel.base`, `height`, `length`, `normalized.losses`, `curb.weight`, `bore`, `highway.mpg`, `city.mpg`, `horsepower` (9 zmiennych)
- wszystkie zmienne numeryczne

Wyniki zostały zilustrowane na rysunku 44. Użyliśmy funkcji `fviz nbclust` [13] z pakietu **factoextra** z parametrem "FUNcluster = kmeans".



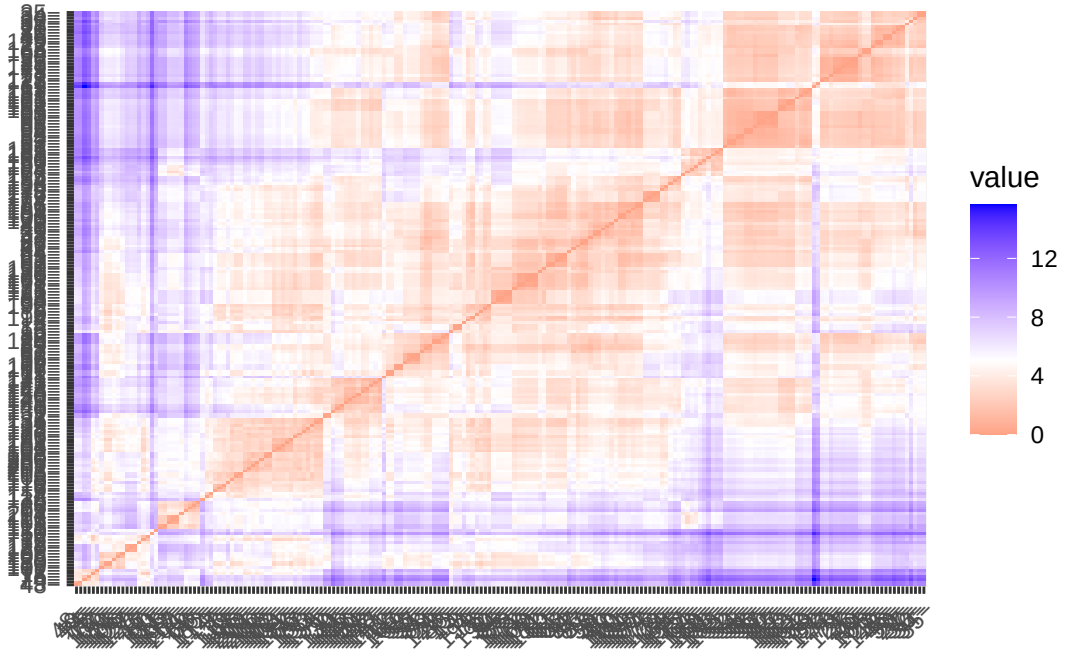

Rysunek 44: Wykres Silhouette dla różnych podzbiorów zmiennych ilościowych i metody k-means

Na podstawie rysunku powyżej możemy wysunąć hipotezę, że dla wszystkich wartości $k$, indeksy Silhouette'a są tym większe, im mniej zmiennych jest branych pod uwagę. Widzimy także, że niezależnie od podzbioru, optymalną liczbą klastrów jest $k = 5$. Ponadto, największe wartości indeksu Silhouette'a dla $k = 2$ lub $k = 3$ osiąga podzbiór zmiennych złożony z `wheel.base` i `price`, a dla $k > 3$ zmienne `horsepower`, `highway.mpg`.

## 4.3 PAM (partition about medoids)

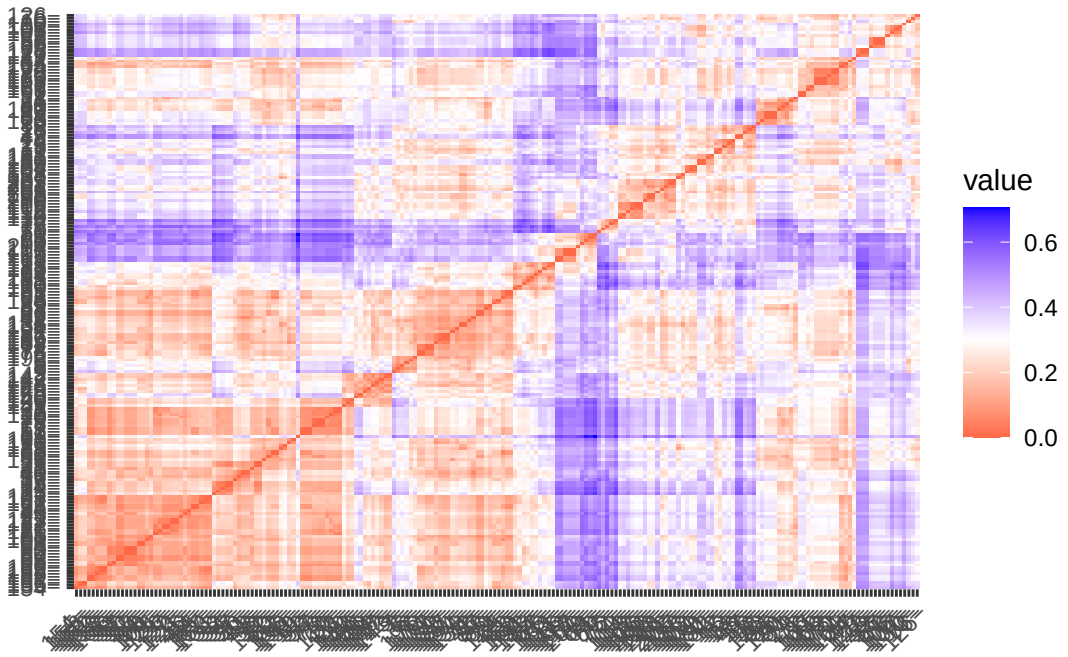
W tej części porównamy metody $k-means$ i $PAM$, ale najpierw kilka słów wstępu o tym drugim algorytmie. Metoda $PAM$ polega na wybraniu $k$ obiektów będących centrami poszczególnych skupień (tzw. medoidami). $PAM$ tak samo jak metoda $k-means$ potrzebuje mieć z góry podaną liczbę $K$ partycji na ile ma być podzieloby zbiór obiektów, ale w przeciwieństwie do $k-means$, na wejściu dostarczamy nie na dane liczbowe, a dowolną macierz odmienności między obiektami. Co z kolei implikuje, że metody $PAM$ możemy użyć nie tylko dla zmiennych ilościowych, ale także dla zmiennych jakościowych, a także danych mieszanego typu. Wadą może być fakt, że $PAM$ ma większą złożoność obliczeniową - proporcjonalną do $n^2$ ($n$-liczba obiektów) niż $k-means$, gdzie złożonosc jest liniowa. Z tego powodu dla dużych $n$ czas obliczeń algorytmu $PAM$ może być problemem, dlatego dla większych zbiorów danych zaleca się używanie algorytmu $CLARA$ (Clustering LARge Applications). Jednakże w przypadku naszych danych, gdzie $n = 205$ (po usunięciu wierszy z `symboling`$= -2$, $n = 202$) będziemy używać zwykłej wersji algorytmu $PAM$.

Jeśli chodzi o implementacje algorytmu $PAM$ w **R**, zaczniemy od przedstawienia macierzy odmienności dla zmiennych ilościowych, oraz dla wszystkich zmiennych. Możemy ja zauważyć na rysunkach 45 i 46. Macierze obliczone są za pomocą funkcji `daisy` [14] z pakietu **cluster**. Domyślną metryką stosowaną w tej funkcji jest metryka euklidesowa. Jednakże jeśli zostaną wykryte zmienne jakościowe, funkcja `daisy` automatycznie stosuje miarę odmienności Gowera [15]. Wizualizacja macierzy została stworzona z użyciem funkcji `fviz dist` z pakietu **factoextra** [16].




Rysunek 45: Macierz odmienności dla zmiennych numerycznych

Poniżej przedstawiona została wizualizacja macierzy odmienności dla wszystkich zmiennych.

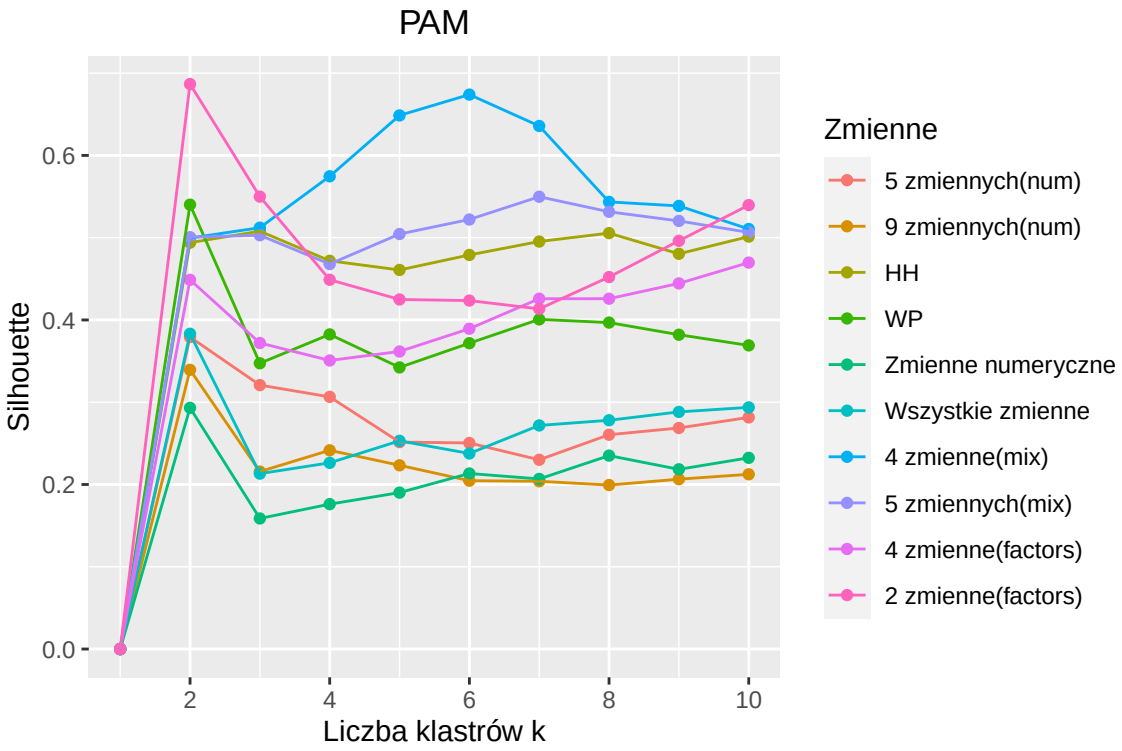



Rysunek 46: Macierz odmienności dla wszystkich zmiennych

Poniżej przedstawiony został wykres dla $PAM$, analogiczny do rysunku 44 dotyczącego metody $k-means$. Zobrazowane zostałow porównanie indeksów Silhouette'a dla różnych wartości $k$ z przedziału $[1, 10]$ oraz dla różnych podzbiorów zmiennych.

- zmienne `horsepower` i `highway.mpg` ($HH$)
- zmienne `wheel.base` i `price` ($WP$)
- zmienne `wheel.base`, `height`, `length`, `normalized.losses`, `curb.weight` (5 zmiennych)
- zmienne `wheel.base`, `height`, `length`, `normalized.losses`, `curb.weight`, `bore`, `highway.mpg`, `city.mpg`, `horsepower` (9 zmiennych)
- wszystkie zmienne numeryczne
- `normalized.losses`, `fuel.system`, `num.of.doors`, `length` (4 zmienne mix)
- `price`, `city.mpg`, `curb.weight`, `body.style`, `drive.wheels` (5 zmiennych mix)
- `make`, `fuel.system`, `num.of.doors`, `body.style` (4 zmienne factors)
- `make`, `num.of.doors` (2 zmienne factors)
- wszystkie zmienne

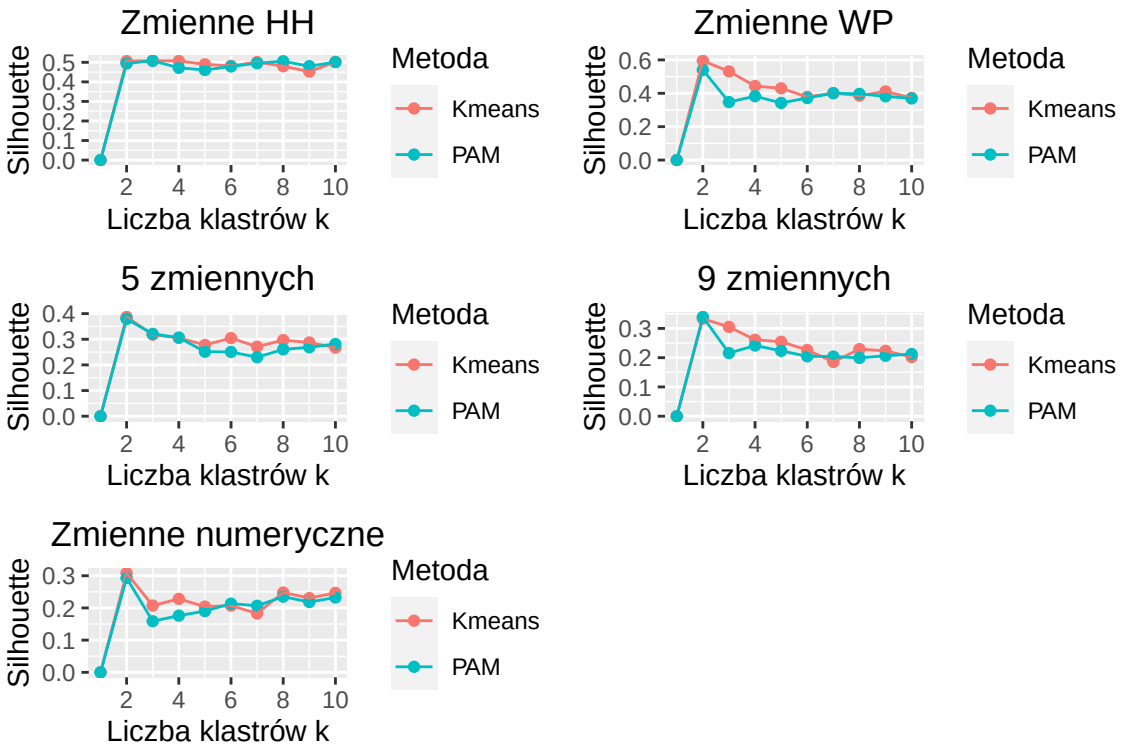
5 podzbiorów, bazujących na zmiennych ilościowych, jest takie samo jak w przypadku metody $k-means$. Kolejne 5 podzbiorów uwzględnia zmienne jakościowe czy dane mieszanego typu. Ponownie użyliśmy funkcji `fviz nbclust` [13] z pakietu **factoextra**, tym razem z parametrem "FUNcluster = cluster::pam", a rezultaty zilustrowane zostały na rysunku 47.




Rysunek 47: Wykres Silhouette dla różnych podzbiorów zmiennych i metody PAM

Po pierwsze, na rysunku powyżej widzimy, że 4 podzbiory zmiennych mają mniejsze indeksy Silhouette'a od pozostałych, są to: "wszystkie zmienne", "zmienne numeryczne", "5 zmiennych(num)" i "9 zmiennych(num)". Po drugie, w tym przypadku widzimy, że nie zawsze $k = 2$ jest optymalną liczbą klastrów. Podzbiory "4 zmienne(mix)", "5 zmiennych(mix)", "4 zmienne(factors)" mają największe wartości indeksów dla odpowiednio $k = 6$, $k - 7$ i $k = 10$. I po trzecie, możemu zauważyć, że największe wartości indeksów dla $k = 2$, $k = 3$ i $k = 10$ są dla podzbioru "2 zmienne (factors)".

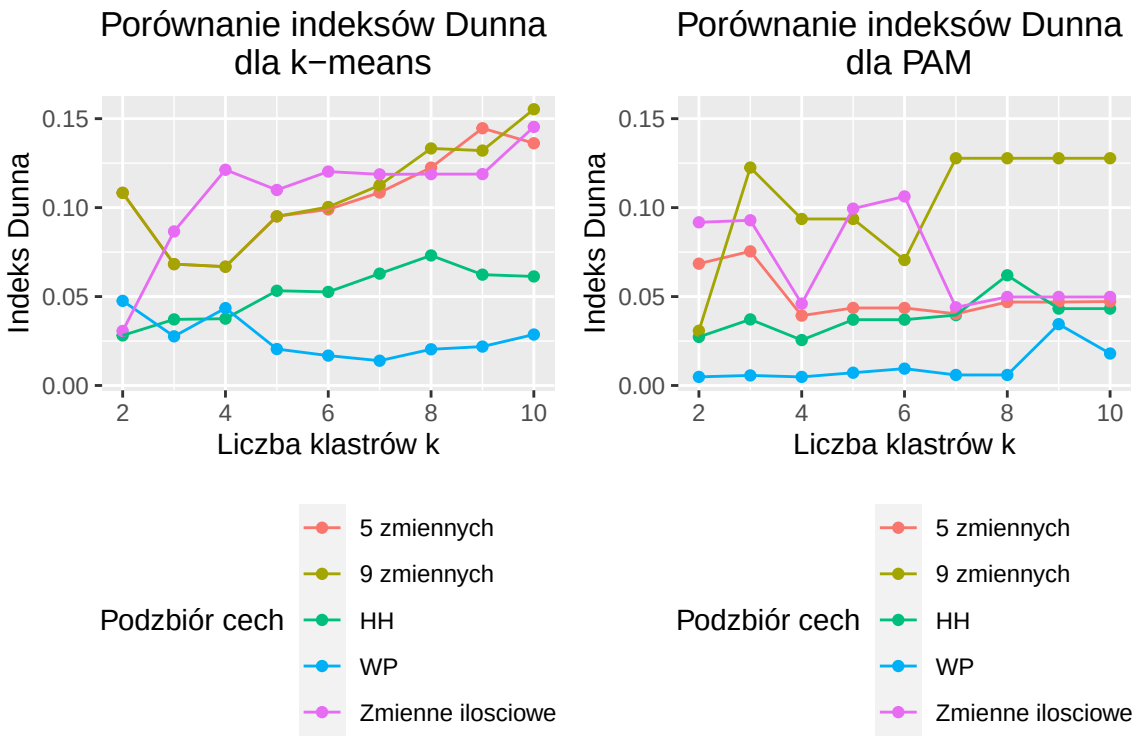
Poniżej na rysunku 48 przedstawione zostało porównanie metod $k - means$ i $PAM$ dla 5 podzbiorów uwzględniających tylko zmienne ilościowe. Oczywiście odpowiednie dane są te same co na rysunkach 44 i 47.



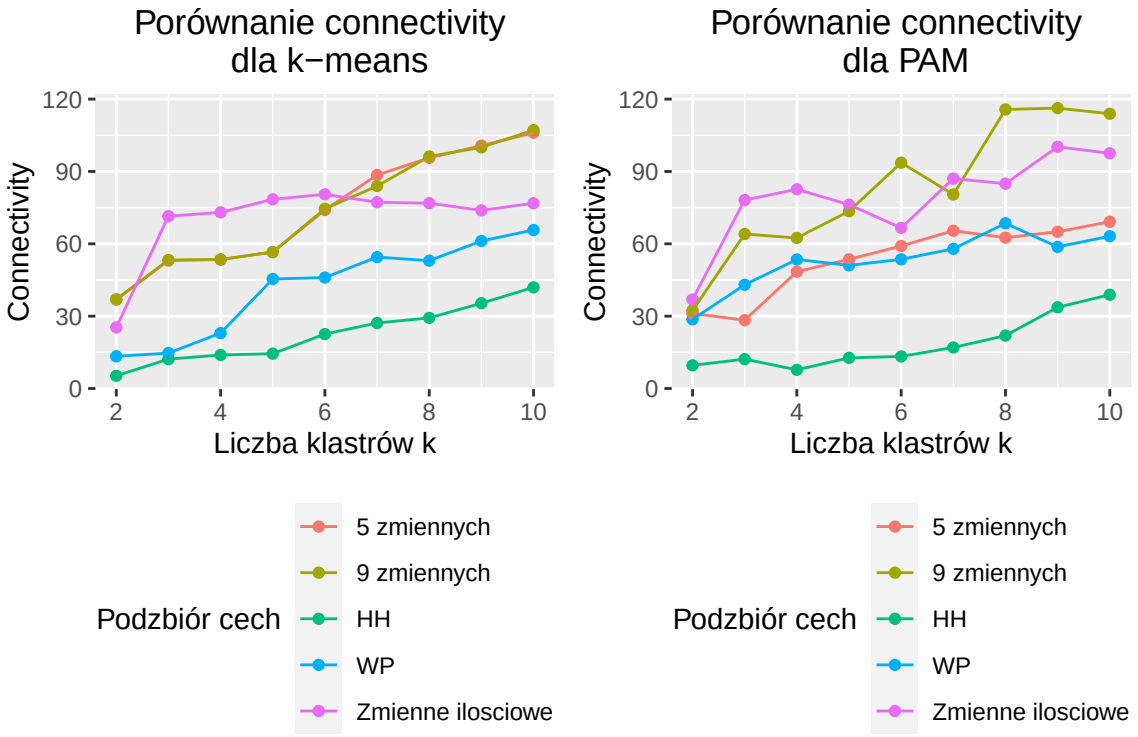

Rysunek 48: Porównanie indeksów Silhouette'a dla zmiennych WP i metody k-means

Na rysunku powyżej, że obie metody zwracają podobne rezultaty jeśli chodzi o indeksy Silhouette'a, niezależnie od wyboru cech czy liczby klastrów. Jednakże widzimy, że dla zdecydowanej większości wypadków lepiej radzi sobie metoda $k - means$ lub obie metody zwracają prawie te same wyniki.

Innymi miarami służącymi do oceny pogrupowania obiektów są wskaźnik Dunn'a [17] i wskaźnik Connectivity. Oba wskaźniki zawsze są nieujemne, różnica polega na tym, że dla wskaźnika Dunn'a im większa wartość, tym lepiej, a dla Connectivity, im mniejsza wartość tym lepiej. W celu obliczeń tych wskaźników posłużyliśmy się funkcjami `dunn` oraz `connectivity` z pakietu **clValid**. Ponadto, w przypadku wskaźnika Connectivity trzeba określić liczbę najbliższych sąsiadów, którą chcemy uwzględnić. W naszym wypadku będzie to $L = 10$. Na rysunkach 49 i 50 przedstawione zostały porównania odpowiednio wskaźników Dunn'a oraz Connectivity dla metod $k - means$ i $PAM$ dla podzbiorów zmiennych numerycznych.




Rysunek 49: Porównanie wartości indeksu Dunna




Rysunek 50: Porównanie wartości wskaźnika connectivity

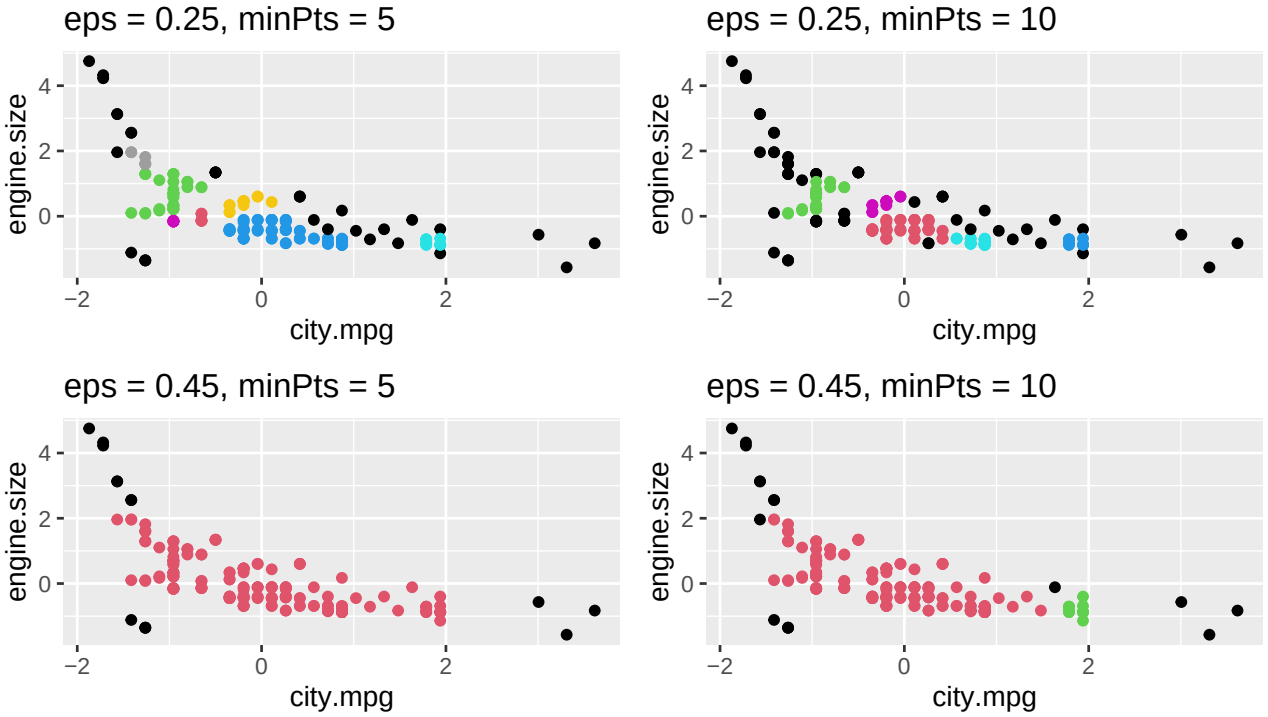
Z rysunków powyżej najważniejszym wnioskiem, niezależnie od interpretacji wartości liczbowych obu wskaźników, które moga się różnić, jest to, że obie metody wypadają kiepsko

biorąc pod uwagę te wskaźniki. Widzimy, że dla wskaźnika Dunn'a największą możliwą wartością, którą otrzymaliśmy jest nieco ponad 0.15, a przypomnijmy, że im większe wartości tym lepiej. Z kolei dla wskaźnika Connectivity otrzymaliśmy duże wartości z minimalną wartością na poziomie ok. 5, a zdecydowana większośc wartości jest większa od 30. Ponadto, przy wskaźniku Connectivity możemy jednoznacznie roztrzygnąć, że najlepiej wypadł podzbiór $HH$ czyli zmienne `horsepower` i `highway.mpg`.

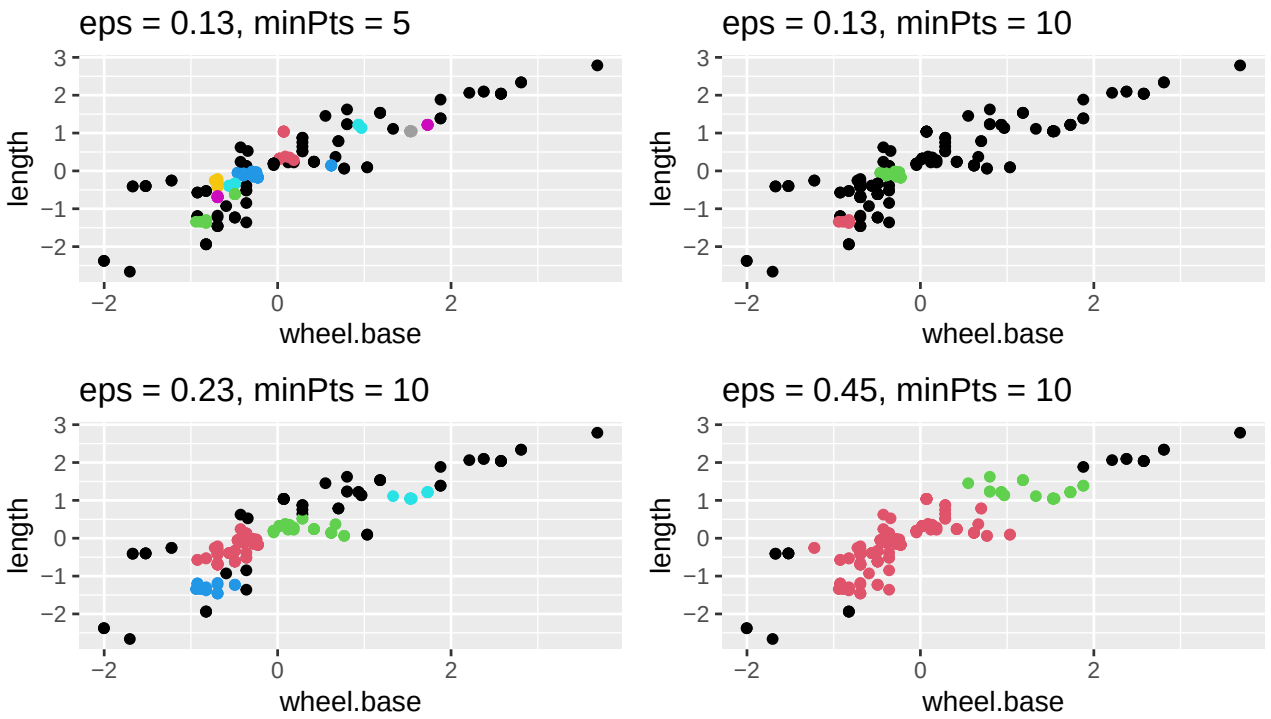
## 4.4 Algorytm DBSCAN

Kolejną metodą klastrowania, której użyliśmy, jest algorytm $DBSCAN$ (ang. Density-Based Spatial Clustering of Applications with Noise). Jest to jedna z metod grupowania gęstościowego, które nieco różnią się od metod grupujących. Przede wszystkim metody grupowania gęstościowego nie potrzebują mieć z góry nadanej liczby klastrów $k$. Poza tym, są bardziej efektywne w przypadku identyfikacji wartości odstających. Warto dodać, że z racji charakteru algorytmu, oczywiście możemy używać tylko zmiennych ilościowych. Więcej o algorytmie $DBSCAN$ tutaj: [18].

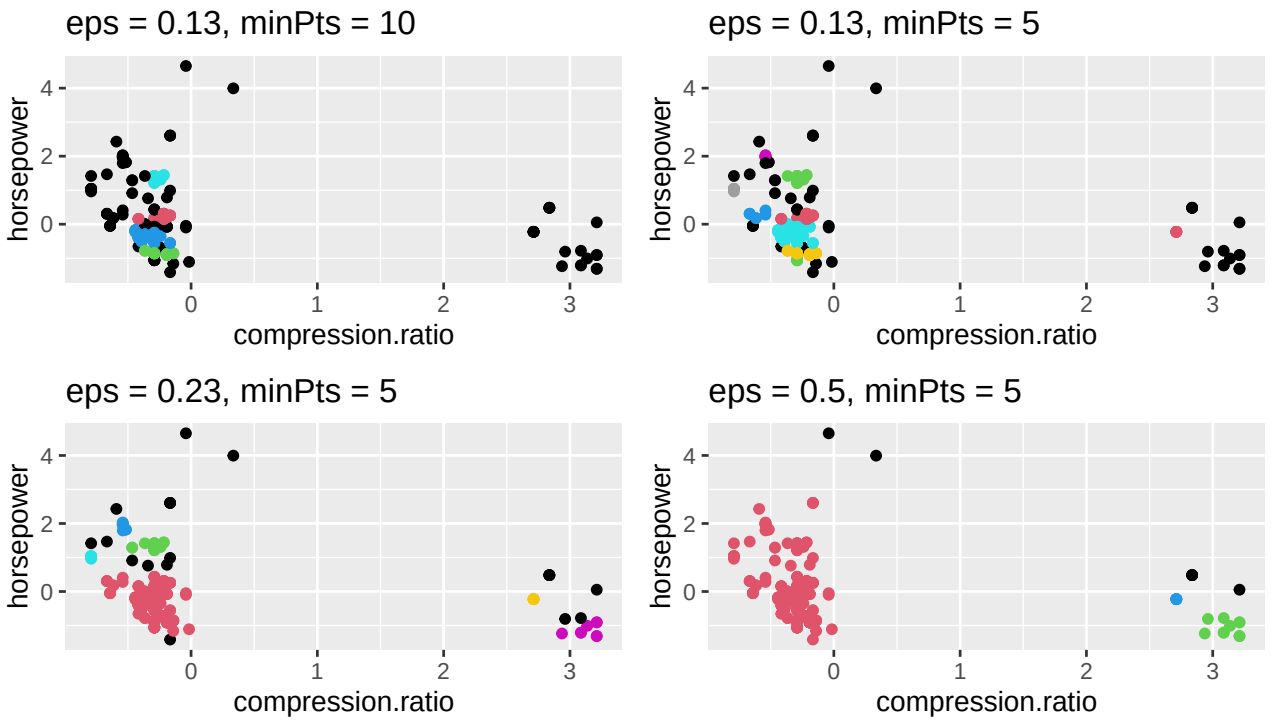
Dla algorytmu $DBSCAN$ musimy ustalić 2 wejściowe wartości: $\epsilon$ oraz $MinPts$. Należy podkreślić, że jedną z wad metody jest duża wrażliwość na wybór $\epsilon$, szczególnosci w przypadku, gdy poszczególne skupienia maja różną gęstość. Do implementacji algorytmu w **R** użyliśmy funkcji `dbscan` z pakietu **dbscan**. Poniżej na rysunkach 51, 52, 53 i 54 przedstawiliśmy wykresy punktowe dla różnych par zmiennych numerycznych z zaimplementowanym algorytmem $DBSCAN$ dla różnych wartości parametrów $\epsilon$ oraz $MinPts$.



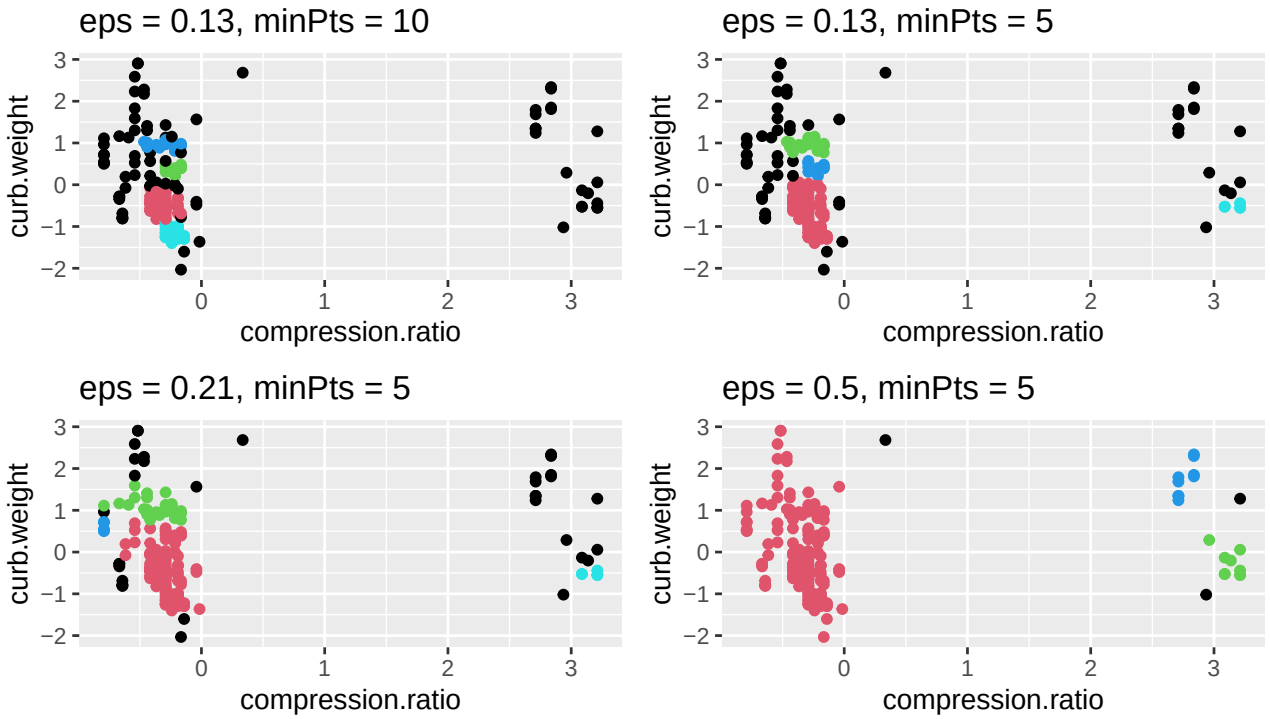

Rysunek 51: Porównanie działania algorytmu DBSCAN dla różnych parametrów eps oraz minPts dla zmiennych city.mpg i enginze.size




Rysunek 52: Porównanie działania algorytmu DBSCAN dla różnych parametrów eps oraz minPts dla zmiennych wheel.base, length




Rysunek 53: Porównanie działania algorytmu DBSCAN dla różnych parametrów eps oraz minPts dla zmiennych compression.ratio, horsepower




Rysunek 54: Porównanie działania algorytmu DBSCAN dla różnych parametrów eps oraz minPts dla zmiennych compression.ratio, curb.weight

Jak możemy zaobserwować na rysunkach powyżej, nie otrzymaliśmy satysfakcjonujących rezultatów dla algorytmu *DBSCAN* dla wybranych par zmiennych, niezależnie od wybranych wartości $\epsilon$ i $minPts$. Prawdopobnie przez bardzo niejednorodną gęstość obserwacji. Oceniając tylko wizualnie, najlepsze wyniki otrzymaliśmy na ostatnim rysunku dla zmiennych `compression.ratio` i `curb.weight` dla $\epsilon = 0.5$ i $minPts = 5$.

## 4.5 Metody hierarchiczne

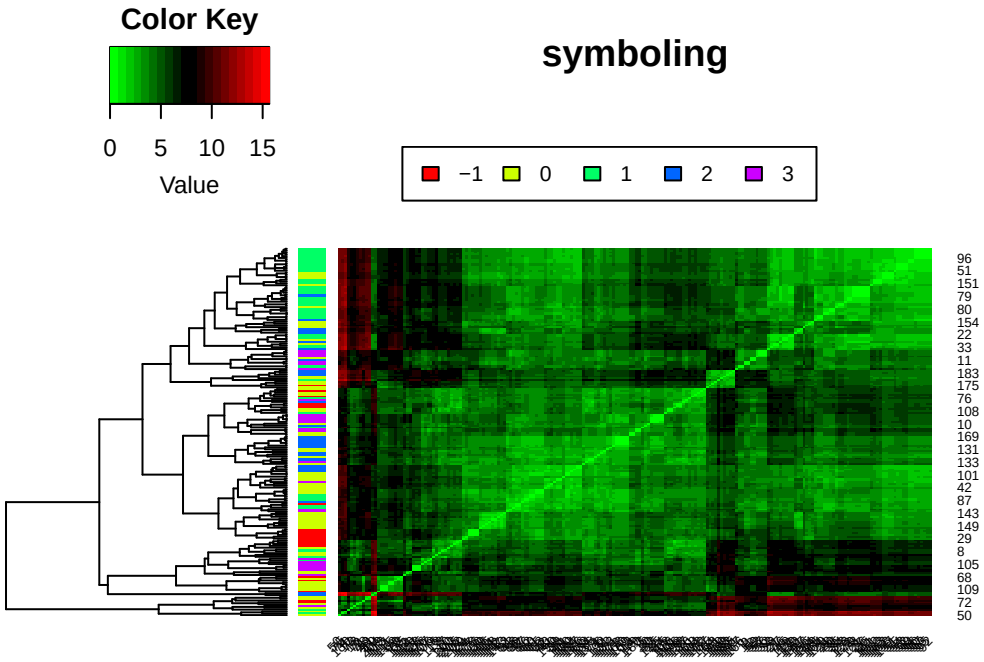
Następnym typem metod których użyliśmy, są metody hierarchiczne. Ich główną ideą jest przedstawienie rezultatów w postaci hierarchii zagnieżdżonych skupień czyli tzw. dendrogramów, podobnie jak w przypadku metod grupowania gęstościowego, tutaj również nie podajemy ustalonej z góry liczb klastrów $k$ na który dzielimy nasz zbiór obiektów, a wyboru dotyczącego tejże liczby dokonujemy po analizie wyników.

### 4.5.1 AGNES

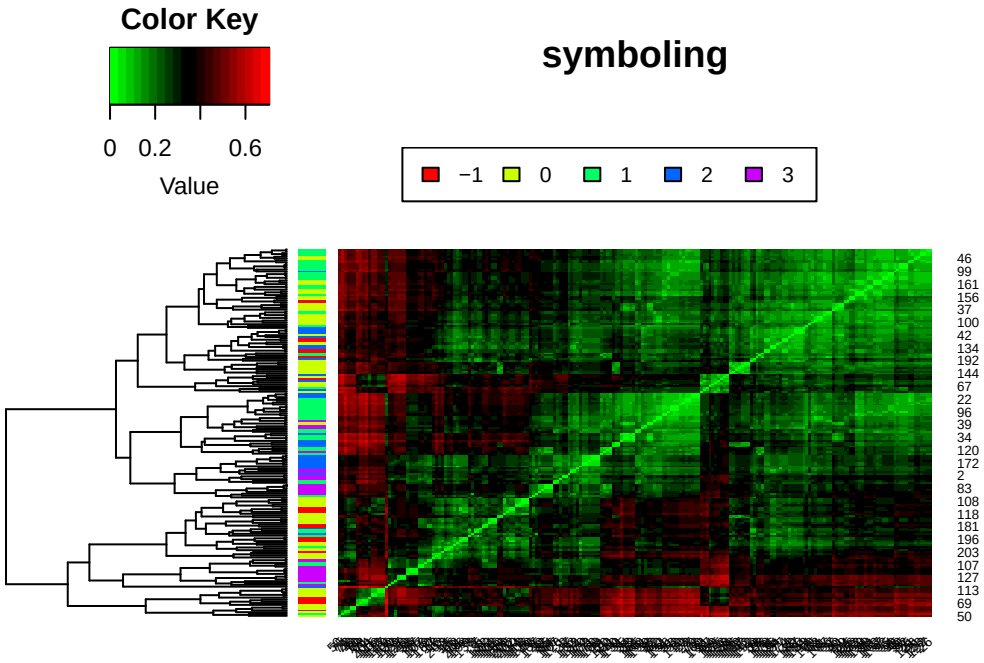
Algorytm *AGNES* jest przykładem metody aglomeracyjnej w których to na początku każdy obiekt stanowi osobne skupienie, a w następnych krokach najbliższe sobie skupienia są łączone, aż do momentu kiedy uzyskamy jedno skupienie zawierające wszystkie obiekty. Na wejściu algorytm *AGNES* otrzymuje macierz odmienności, co z kolei sprawia, że możemy wziąć pod uwagę zarówno zmienne ilościowe jak i jakościowe, oraz oczywiście dane mieszanego typu.

I właśnie macierzami odmienności zajmiemy się na samym początku, a konkretnie ich wizualizacją. Co prawda, już wcześniej przy metodzie *PAM* zostały dodane takie rysunki, ale tutaj do naszych heatmap dodamy także dendrogramy oraz faktyczne etykietki zmiennej objaśnianej `symboling` (warto dodać, że korzystamy w końcu z naszej kosntrukcji zmiennej `symboling`

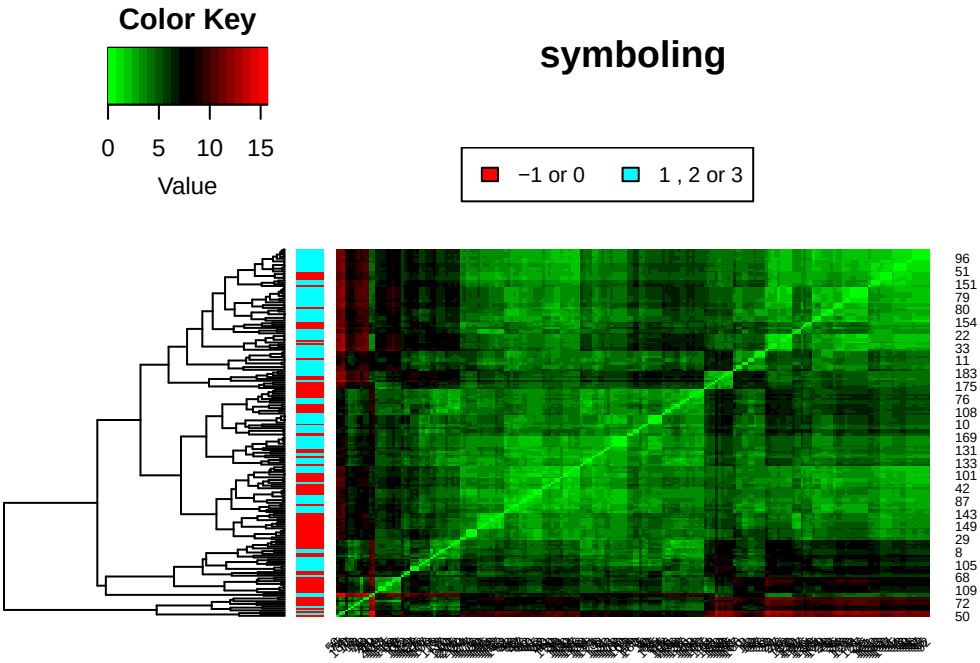
z 2 kategoriami). Przedstawione zostały one na rysunkach 55, 56, 57 i 58. Do ich stworzenia użyliśmy funkcji `heatmap.2` [19] z pakietu **gplots**.



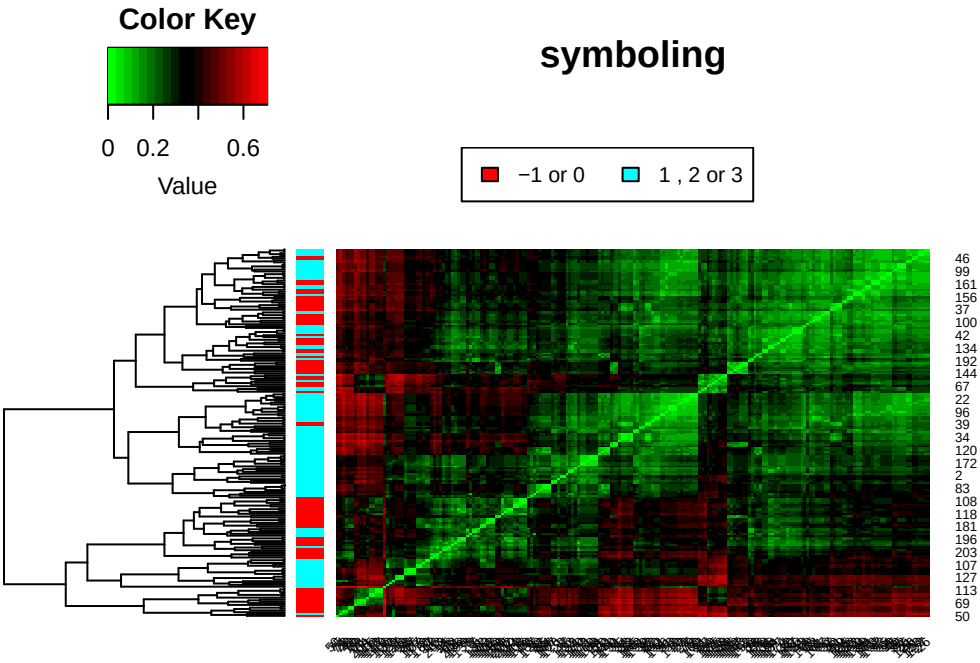

Rysunek 55: Heatmapa dla macierzy odmienności uwzględniającej zmienne numeryczne




Rysunek 56: Heatmapa dla macierzy odmienności uwzględniającej wszystkie zmienne




Rysunek 57: Heatmapa dla macierzy odmienności uwzględniającej zmienne numeryczne (zmienna symboling sprowadzona do 2 kategorii)




Rysunek 58: Heatmapa dla macierzy odmienności uwzględniającej wszystkie zmienne (zmienna symboling sprowadzona do 2 kategorii)

Na podstawie powyższych rysunków, możemy stwierdzić, że otrzymane hierachie zagnieżdżonych skupień kiepsko odwzorowują rzeczywiste kategorie zmiennej `symboling`.

Ważną informacją przy ocenie jakości otrzymanej struktury jest współczynnik aglomeracyjny ($AC$). Im bliżej wartość jest 1, tym bardziej istotny jest otrzymany podział. Dla każdego

obiektu $i$ wyznaczamy $m(i)$, czyli odmienność do pierwszego skupienia, z którym została połączona, a następnie dzielimy ją przed odmienność od skupienia dołączonego w ostatnim kroku algorytmu. Wartość $AC$ to średnia ze wszystkich wartości $1 - m(i)$

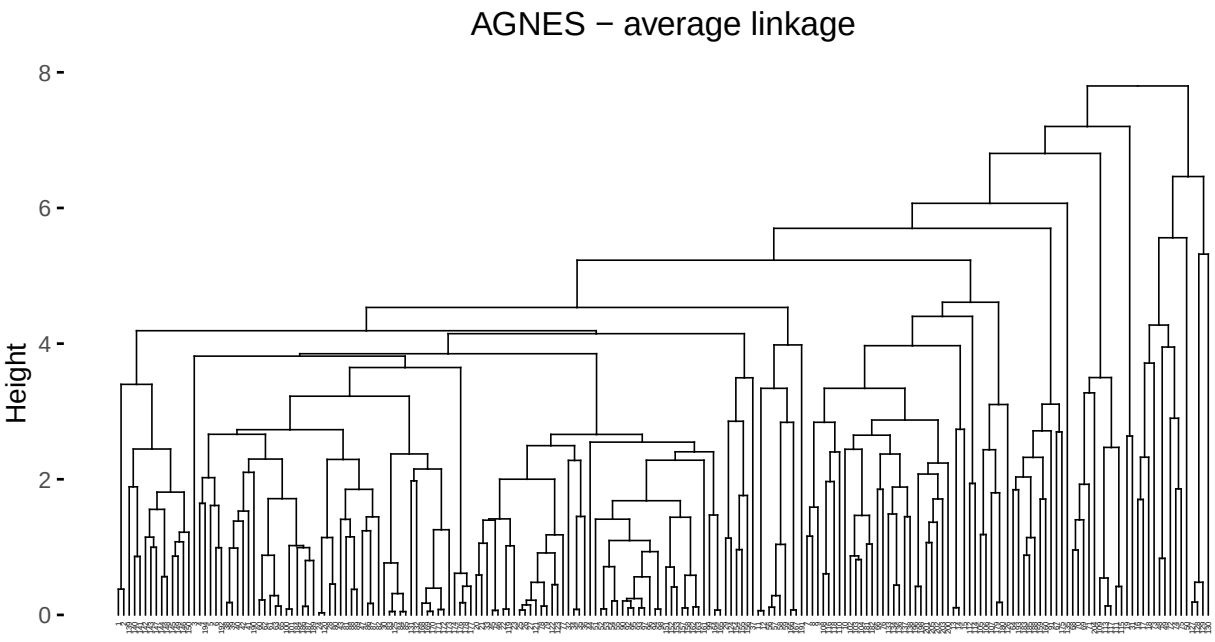
Bardzo ważną sprawą dotyczącą metody $AGNES$ jest wybór metody łączenia klastrów. Istnieją 3 różne typy takich łączeń:

- Odległość najbliższego sąsiada (ang. simple linkage)
- Odległość najdalszego sąsiada (ang. complete linkage)
- Odległość średnia (ang. average linkage)

Ale nie są to jedyne metody. Najpopularniejszą z pozostałych jest prawdopodobnie metoda Warda. Więcej o metodach łączenia klastrów możemy znaleźć tutaj [20].

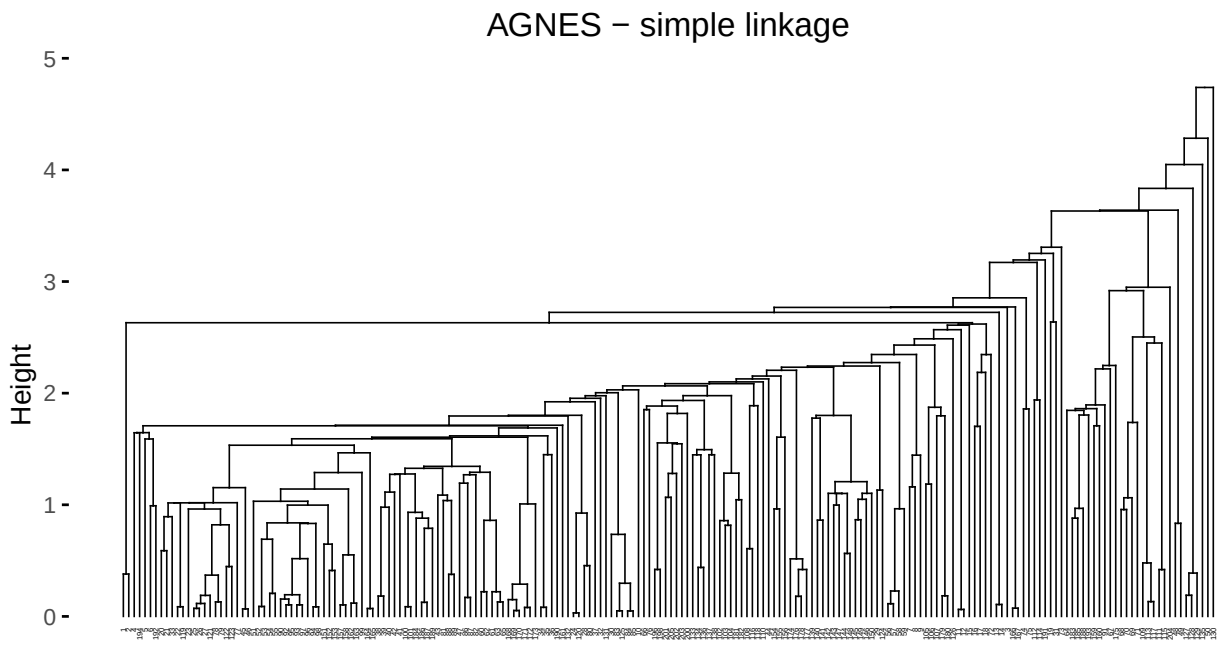
Żeby zwizualizować jak ważny jest wybór metody łączenia klastrów poniżej na rysunkach 59, 60, 61 przedstawione zostały dentrogramy metody $AGNES$ uzytej na zmiennych ilościowych dla odpowiednio `average linkage`, `simple linkage` i `complete linkage`. Do ich stworzenia użyliśmy funckji `agnes` [21] z pakietu **cluster**, a następnie funkcji `fviz dend` [**?**] z pakietu **factoextra** Oprócz tego dla każdego z typów łączeń wyświetlony został także współczynnik $AC$.

```
## [1] "Współczynnik AC dla metody łączenia average linkage wynosi: 0.87"
```



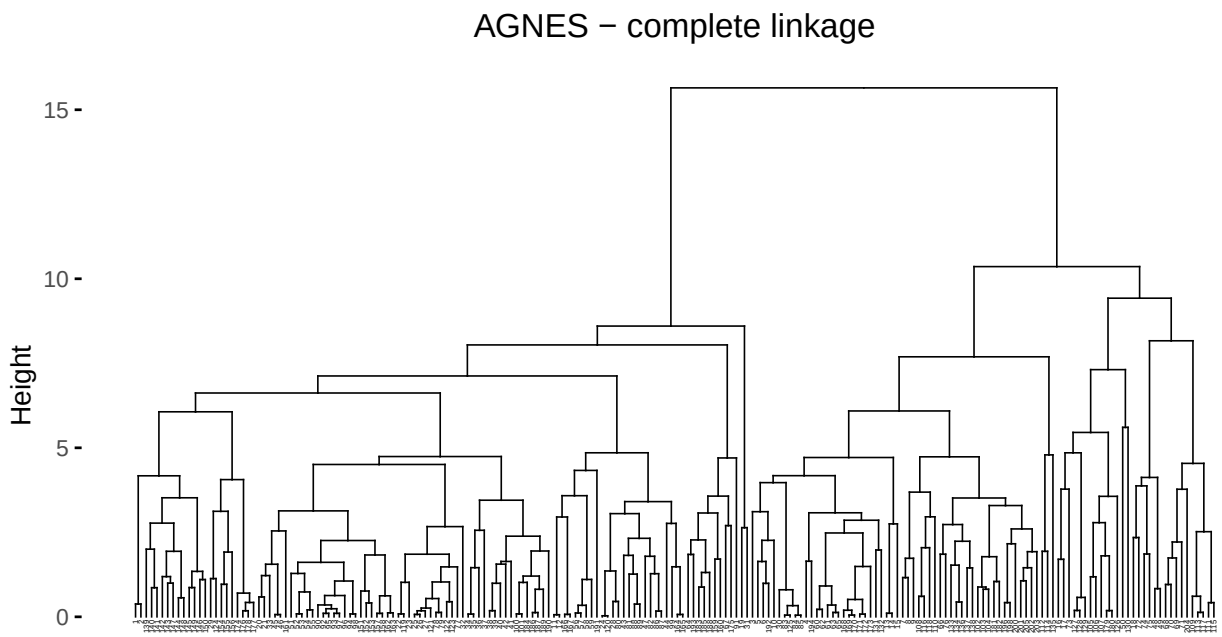

Rysunek 59: dendrogram dla metody AGNES dla zmiennych ilościowych z wykorzystaniem average linkage

```
## [1] "Współczynnik AC dla metody łączenia single linkage wynosi: 0.81"
```



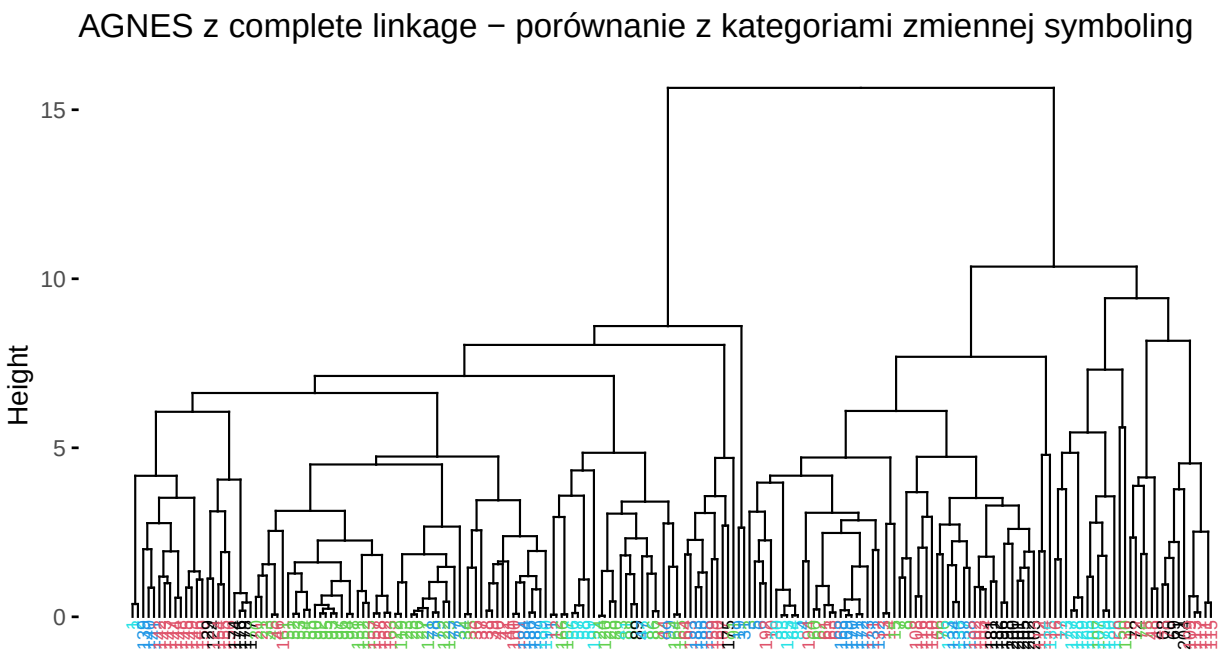

Rysunek 60: dendrogram dla metody AGNES dla zmiennych ilościowych z wykorzystaniem single linkage

```
## [1] "Współczynnik AC dla metody łączenia complete linkage wynosi: 0.93"
```

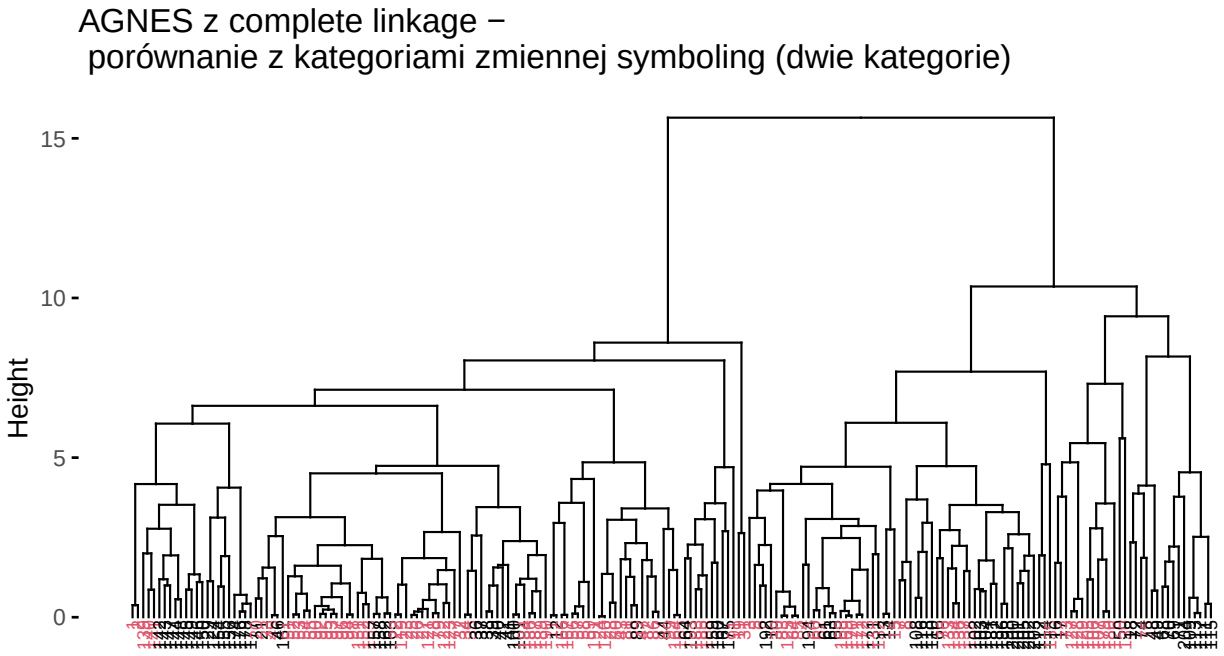



Rysunek 61: dendrogram dla metody AGNES dla zmiennych ilościowych z wykorzystaniem complete linkage

Z 3 powyższych rysunków, możemy wywnioskować, że najbardziej sensownym podziałem jest ten z pomocą metody łączenie `complete linkage`. Potwierdza to też najwyższa wartość współczynnika $AC$, właśnie dla tej metody, który jest równy 0.93. I właśnie dendrogramów przy użyciu tej metody łączenia uzyjemy do porównania ze zmiennymi jakościowymi. Poniżej na rysunkach 62 i 63 przedstawione są dendrogramy dla metody AGNES (*completelinkage*) na podstawie zmiennych ilościowych z zaznaczonymi etykietkami zmiennej `symboling`.



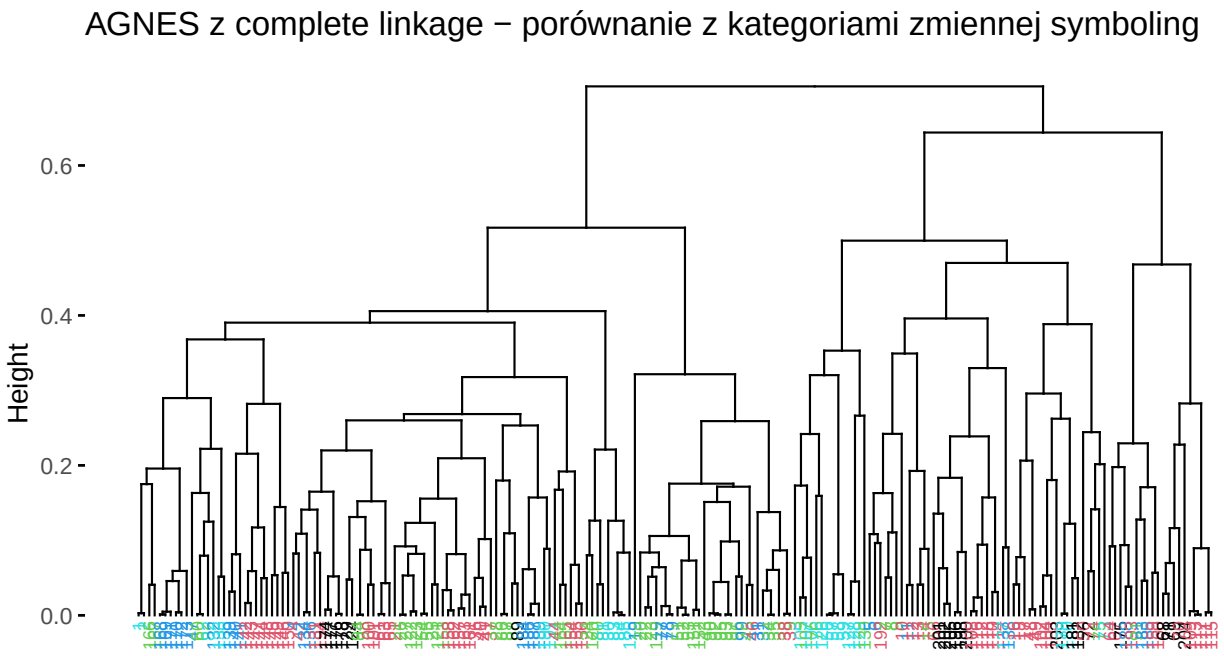

Rysunek 62: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej symboling



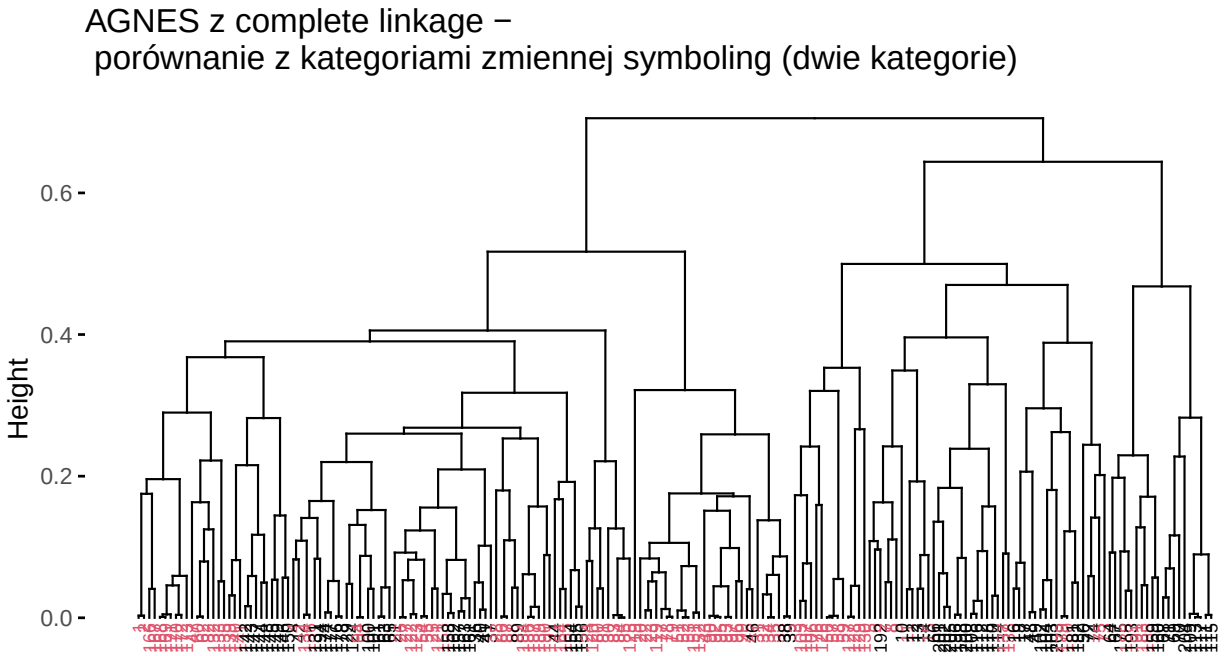

Rysunek 63: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej symboling (podział na 2 kategorie)

Poniżej na rysunkach 64 i 65 przedstawione są dendrogramy dla metody AGNES (*completelinkage*), tym razem na podstawie wszystkich zmiennych, także jakościowych (oczywiście oprócz zmiennej

objaśnianej `symboling`) z zaznaczonymi etykietkami zmiennej `symboling`.



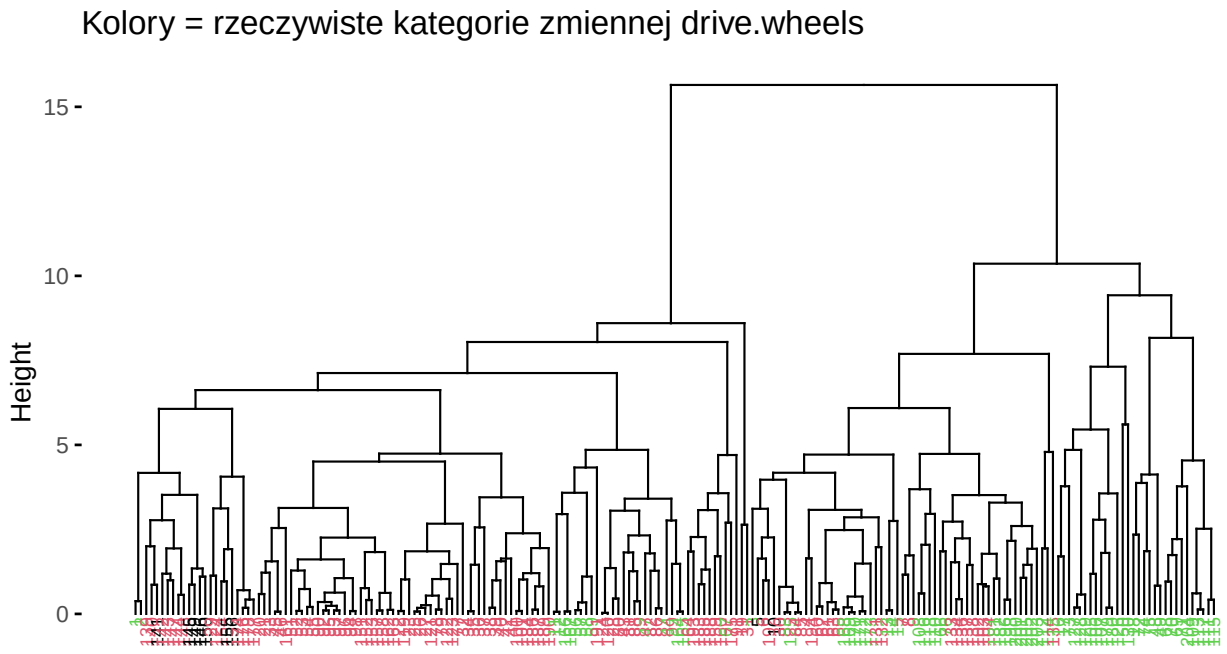

Rysunek 64: dendrogram dla metody AGNES dla wszystkich zmiennych z zaznaczeniem różnych kategorii zmiennej symboling



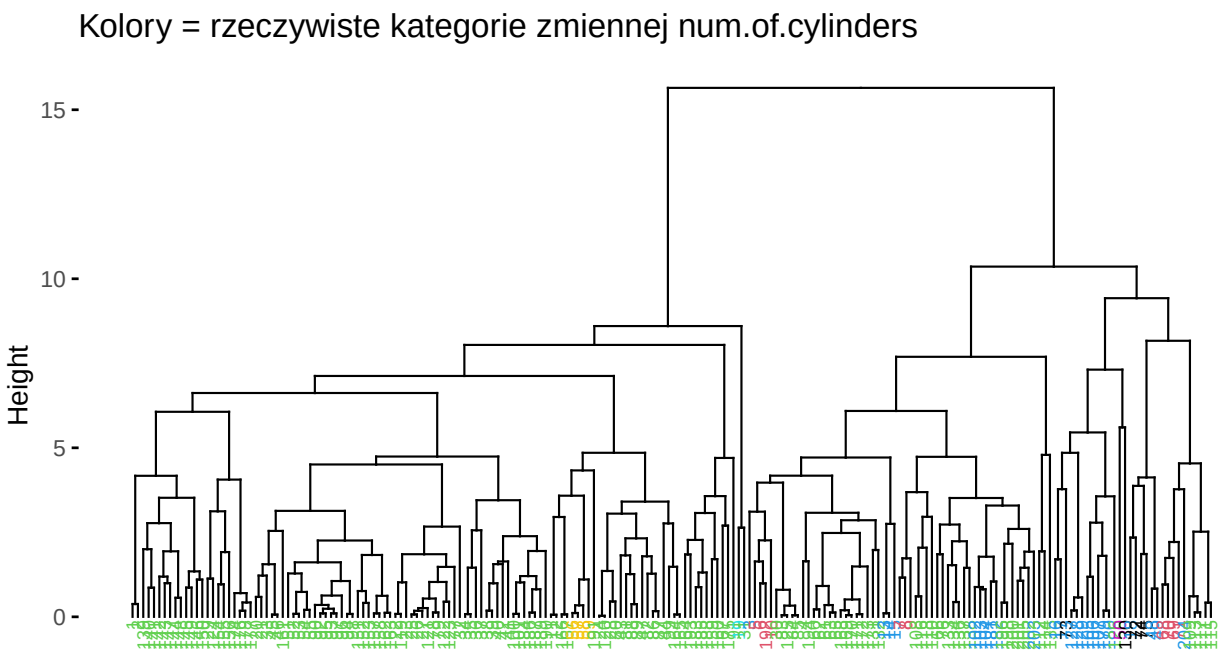

Rysunek 65: dendrogram dla metody AGNES dla wszystkich zmiennych z zaznaczeniem różnych kategorii zmiennej symboling(podział na 2 kategorie)

Na 4 powyższych rysunkach możemy zaobserwować, że niezależnie od użycia wszystkich zmiennych czy tylko ilościowych, zagnieżdżenia uzyskiwane metodą *AGNES* nie odpowiadają wyraźnie jakiemuś podziałowi związanemu ze zmienną `symboling`

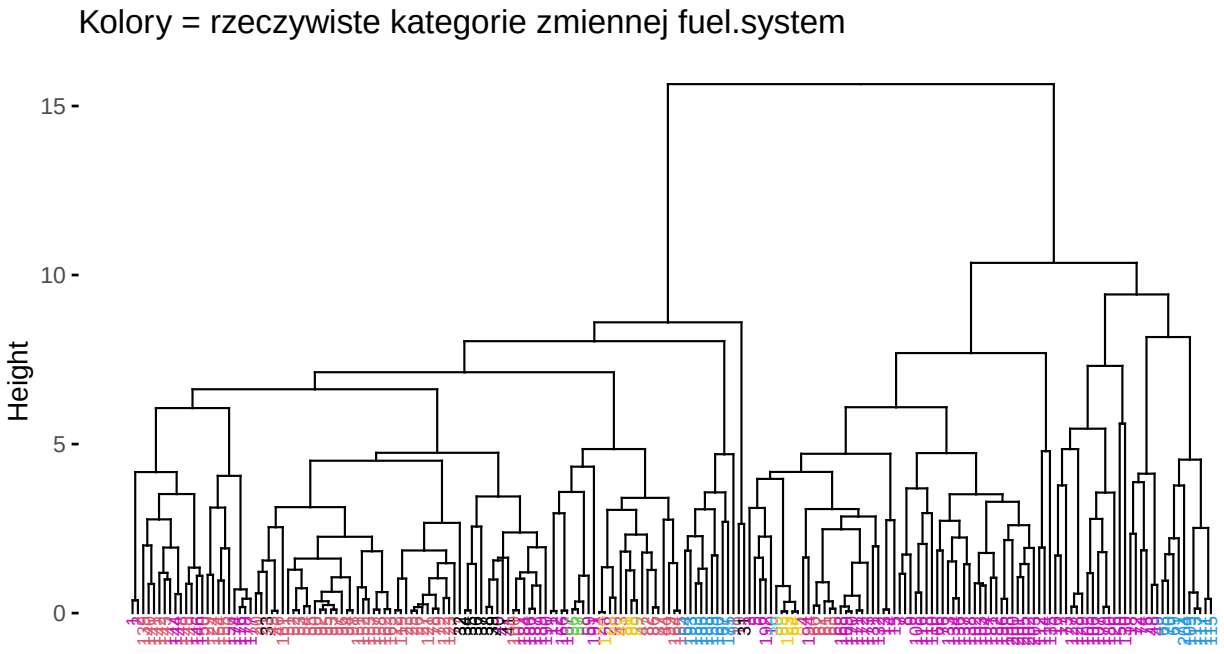
Poniżej na rysunkach 66, 67 i 68 przedstawione zostały dendrogramy uzyskane metodą *AGNES* dla metody łączenia klastrów *completelinkage*, na podstawie zmiennych ilościowych, z porównaniem z różnymi innymi zmiennymi jakościowymi: `drive.wheels`, `num.of.cylinders` i `fuel.system`.




Rysunek 66: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej drive.wheels




Rysunek 67: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej num.of.cylinders



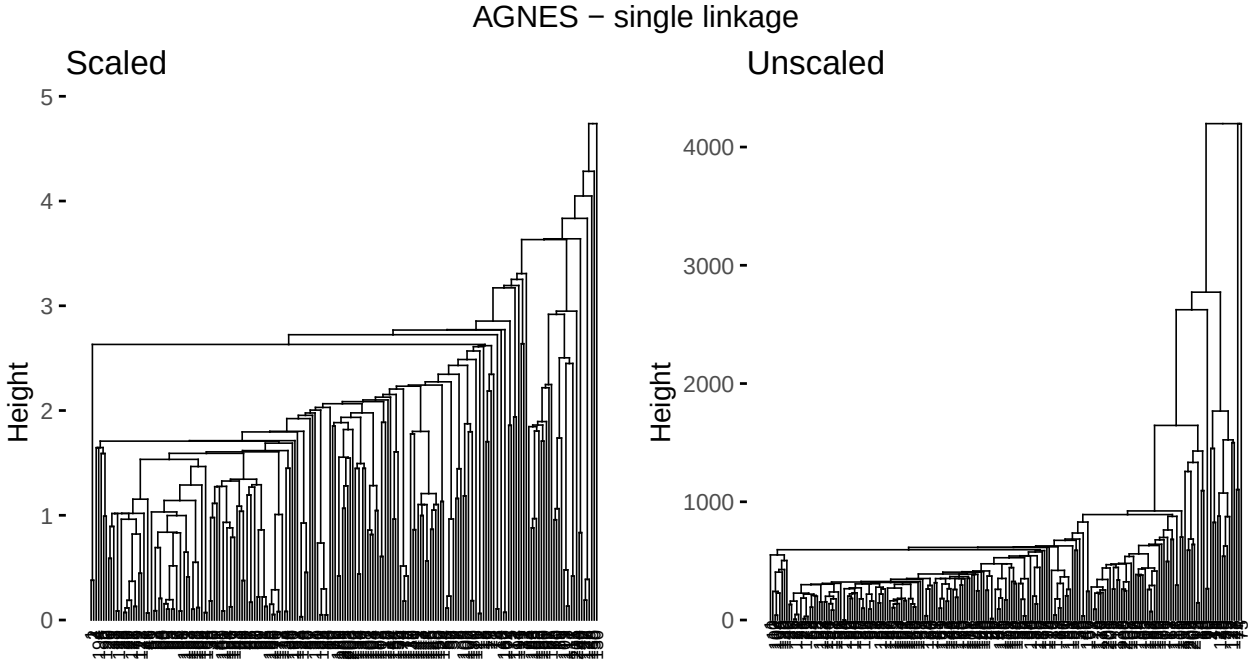

Rysunek 68: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej fuel.system

Na powyższych rysunkach widzimy, że etykietki każdej ze zmiennych `drive.wheels`, `num.of.cylinders` i `fuel.system` w jakiś sposób lepiej odpowiadają strukturze klastrów niż dla zmiennej `symboling`

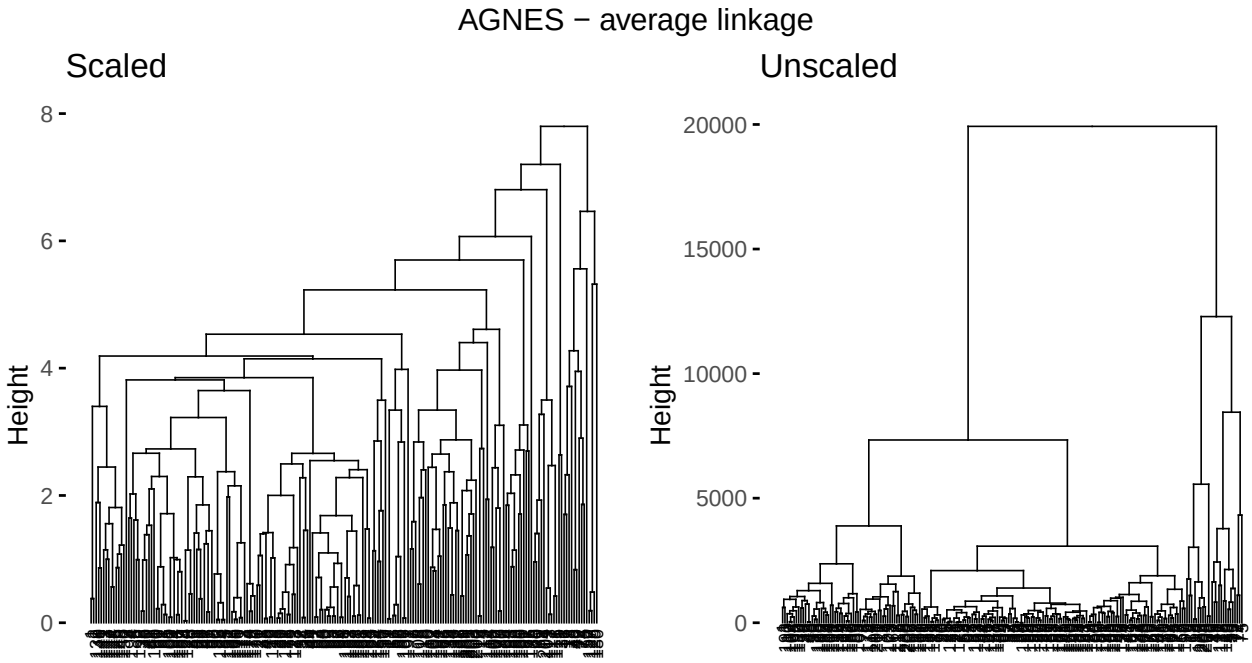
np. dla zmiennej `num.of.cylinders` zdecydowana część lewej (z dwóch) gałęzi dendrogramu należy do kategorii oznaczonej kolorem zielonnym, z kolei da zmiennej `fuel.system`, zdecydowana większość prawej gałezi dendrogramu należy do etykietki oznaczonej kolorem różowym.

Stworzone zostały one także dendrogramy porównujące z kategoriami pozostałych zmiennych jakościowych. Wykresy te umieszczone zostały w sekcji Dodatek.

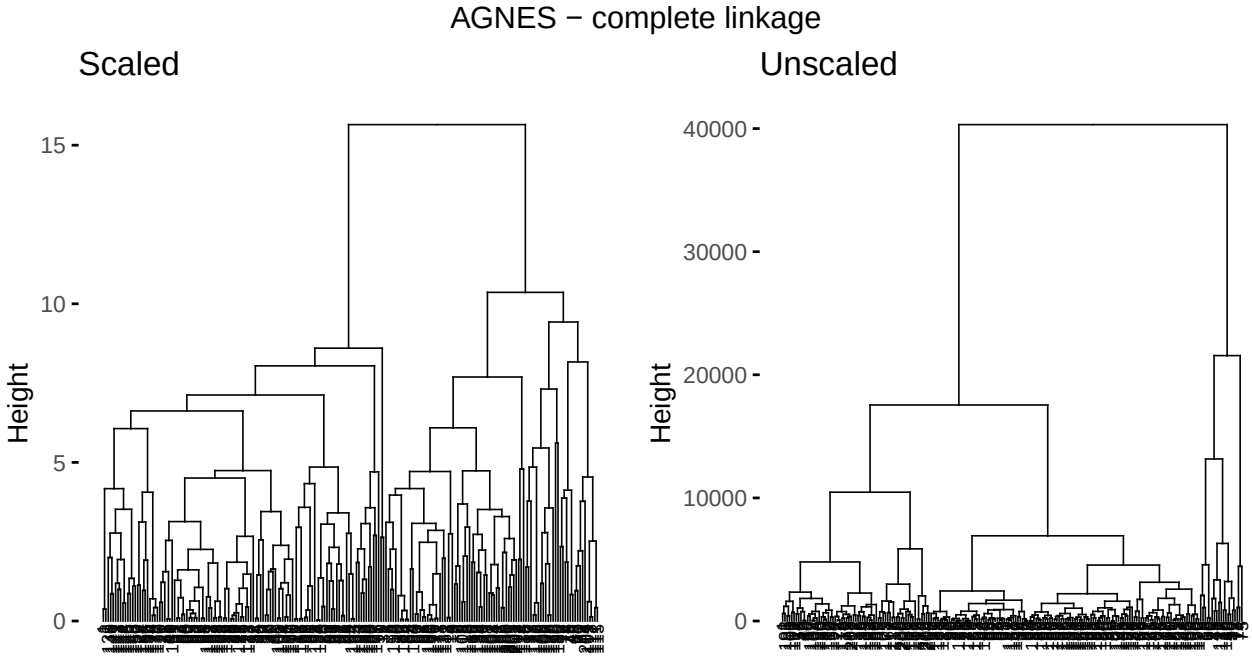
Dodatkowo, warto przypomnieć fakt, że wszystkie zmienne numeryczne były standaryzowane przed użyciem. Dlaczego tak zrobiliśmy? Poniżej odpowiedź na rysunkach 69, 70 i 71. Przedstawione są tam porównawcze dendrogramy między użyciem ustandaryzowanych i oryginalnych danych liczbowych dla każdej z trzech wcześniej używanych przez nas metod łączenia.




Rysunek 69: dendrogram dla metody AGNES dla zmiennych ilościowych z wykorzystaniem complete linkage




Rysunek 70: dendrogram dla metody AGNES dla zmiennych ilościowych z wykorzystaniem complete linkage




Rysunek 71: dendrogram dla metody AGNES dla zmiennych ilościowych z wykorzystaniem complete linkage

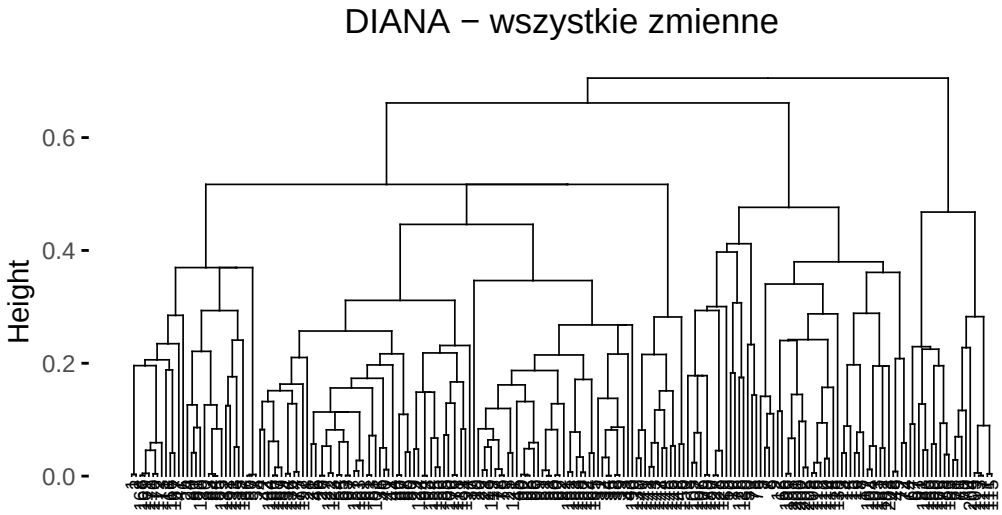
Widzimy na powyższych 3 rysunków, że dendrogramy uzyskane na podstawie macierzy odmienności uzsykanych na bazie danych niestandaryzowanych mają zupełnie inne struktury.

Bez zastosowania standaryzacji mamy oczywiście inną skalę dla odległości, co przekłada się na wygląd dendrogramów.

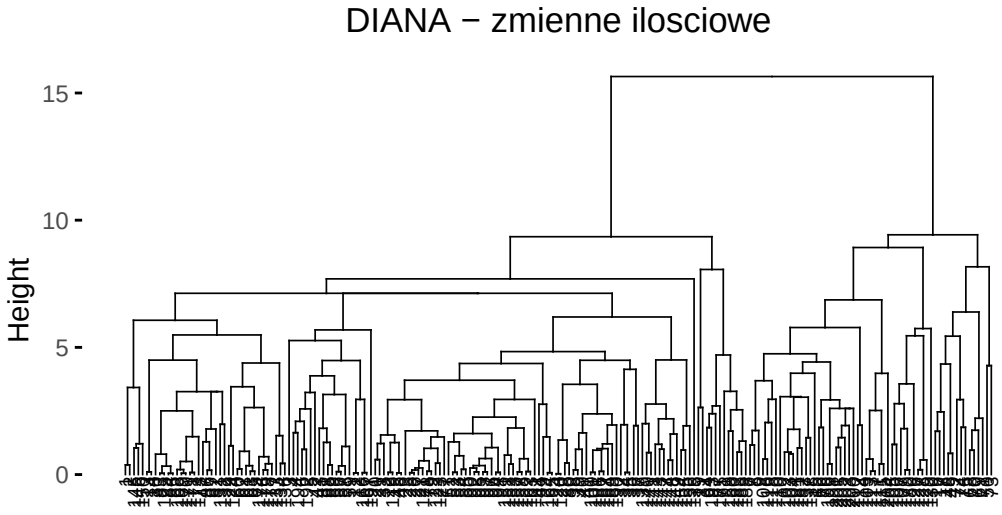
### 4.5.2 DIANA

Kolejną metodą, której użyjemy jest $DIANA$ (ang. DIviseve ANAlysis). Jest to przykład metody dzielącej, która działają niejako "odwrotnie" niż metody aglomeracyjne tzn. na początku wszystkie obiekty tworzą jedno duże skupienie, które jest potem dzielone, tak aby otrzymać jednorodne klastry. $DIANA$, podobnie jak $AGNES$ potrzebuje na wejściu tylko macierz odmienności, więc oczywiście możemy używać zmiennych jakościowych jak i danych mieszanego typu.

Na początek porównamy metodę $DIANA$ dla wszystkich zmiennych i tylko dla zmiennych ilościowych. Na rysunkach 72 i 73 przedstawione zostały takie dendrogramy.



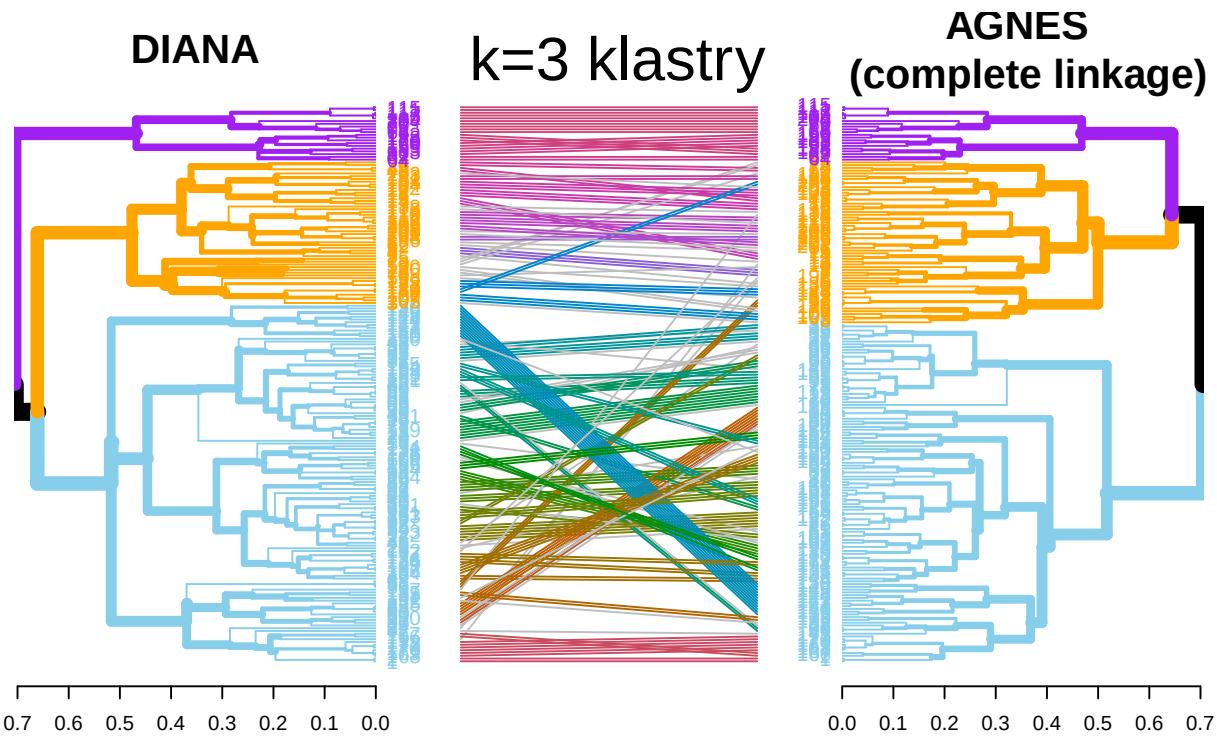

Rysunek 72: dendrogram dla metody DIANA dla wszystkich zmiennych



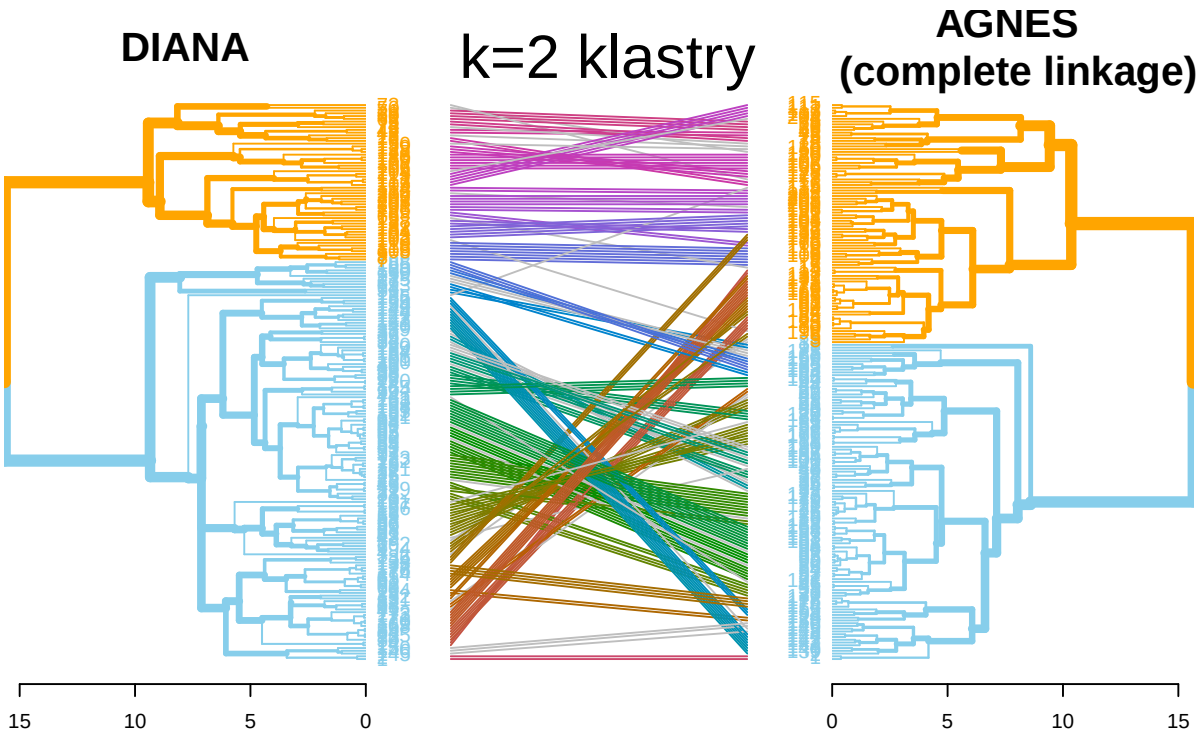

Rysunek 73: dendrogram dla metody DIANA dla zmiennych ilościowych

Widzimy na dwóch powyższych rysunkach, że otrzymane dendogramy znacząco się różnią strukturą. Najwżniejszą obserwacją jest fakt, że w przypadku użycia wszystkich zmiennych, najbardziej wyraźny jest podział na $k = 3$ klastry, natomiast dla zmiennych numerycznych są $k = 2$ klastry. Możemy także zauważyć, że wobu przypadkach, metoda $DIANA$ zwraca rezultaty podobne do algorytmu $AGNES$ z metodą łączenia klastrów complete linkage.

Ciekawym graficznym sposobem, żeby porównać 2 dendrogramy jest tzw. tanglegram, który umieszcza 2 dendrogramy w pozycji na przeciwko siebie, a na środku umieszcza linie łączące te same obiekty między dwoma dendrogramami. Poniżej na rysunkach 74 i 75 umieszczone zostały tanglegramy dla metody $DIANA$ i $AGNES$ (complete linkage) dla wszystkich zmiennych i dla zmiennych numerycznych odpowiednio. Co więcej, na owich tanglegramach zilustrowany został podziała na klastry, gdzie $k = 3$ w przypadku wszystkich zmiennych i $k = 2$ dla zmiennych numerycznych. Tanglegramy zostały stworzone za pomocą funkcji `tanglegram` z pakietu **dendextend**.




Rysunek 74: Tanglegram dla metod DIANA i AGNES dla wszystkich zmiennych

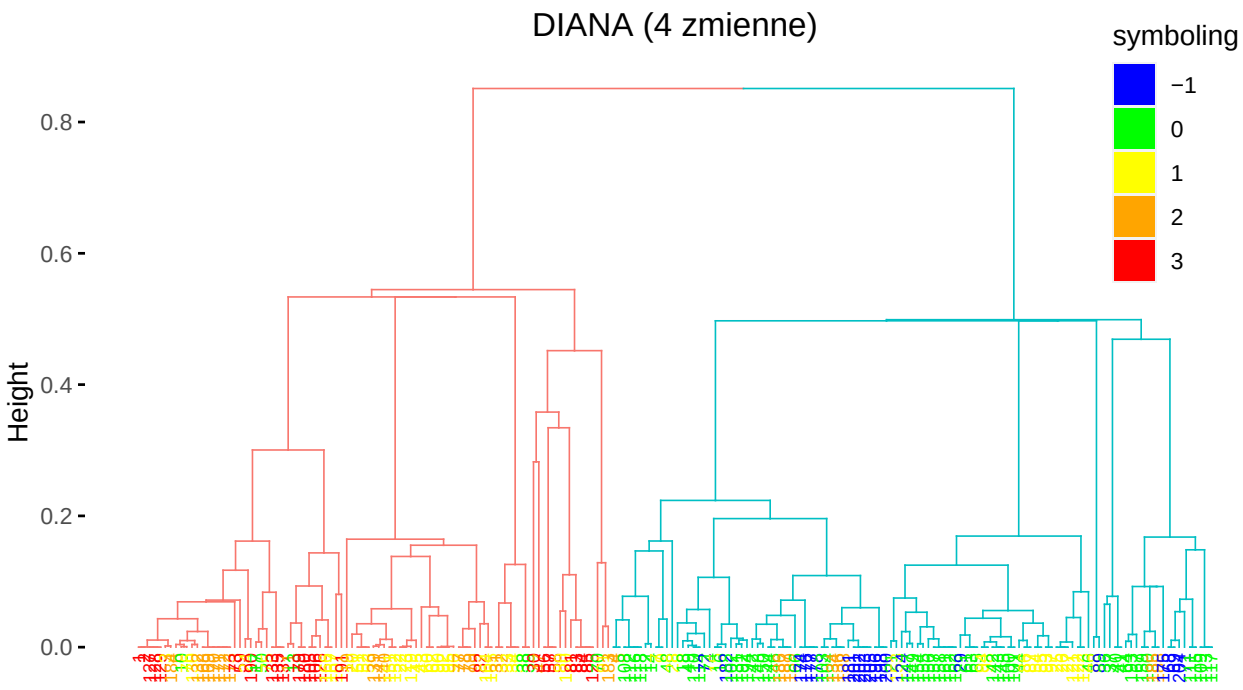



Rysunek 75: Tanglegram dla metod DIANA i AGNES dla zmiennych ilościowych

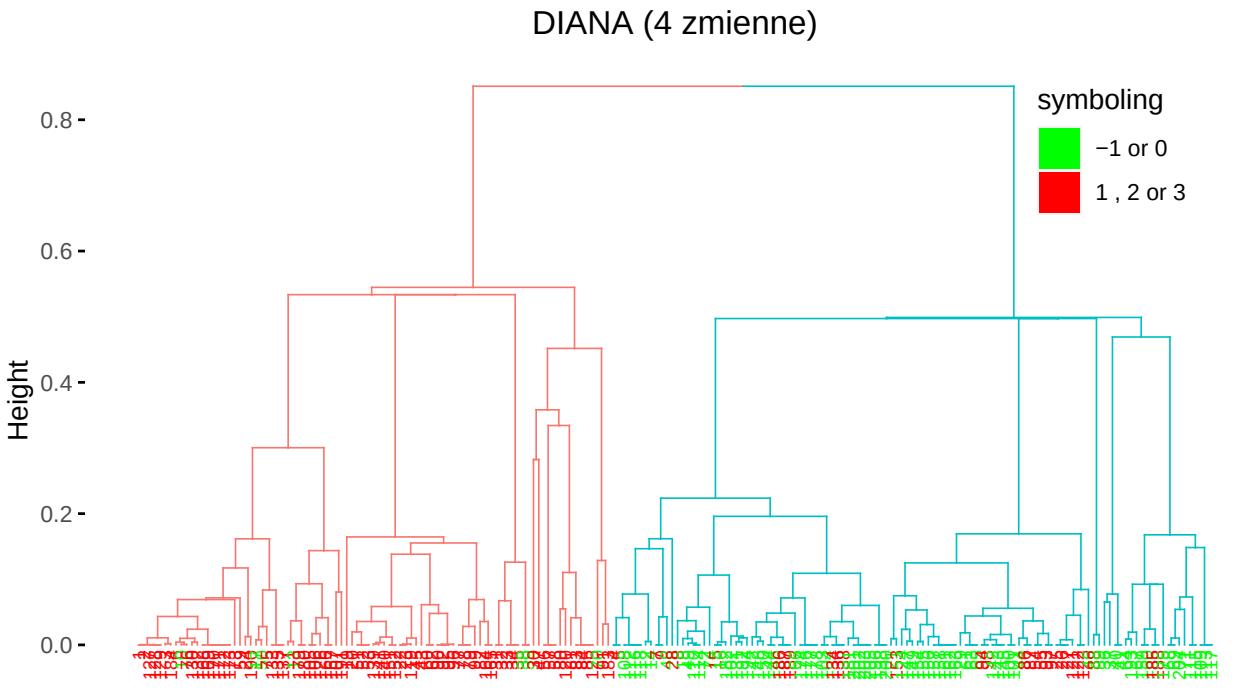
Jak wyżymy powyżej, dla przypadku wszystkich zmiennych dendrogramy uzyskane metodami *DIANA* i *AGNES* są bliskie identyczności. Widzimy, że linie na środku tylko w pojedynczych przypadkach wychodzą z dendrogramu po lewej stronie z klastru o jednym kolorze, do dendrogramu po prawej stronie do klastra o kolorze innym. Dodatkowo, w obu przypadkach widzimy, że podział na $k = 3$ klastry jest jak najbardziej optymalny.

Nieco inaczej jest w przypadku zmiennych ilościowych, gdzie różnica między dendrogramami jest znacznie większa, ale wciąż otrzymane rezultaty możemy określić jako bardzo podobne.

W następnej części przeprowadzone zostały badania z różnymi niestandarowanymi podzbiorami cech, wybiórczo przez nas wybranymi. Poniżej na rysunkach 76, 77 przedstawione zostały dendrogramy uzyskane metodą *DIANA* na bazie 4 następujących zmiennych: `normalized.losses`, `fuel.system`, `num.of.doors`, `length`. Gałęzie dendrogramu zostały pokolorowane przyjmując podział na $k = 2$ klastry, a na dole etykietki obiektów zostały pokolorowane zgodnie z etykietkami zmiennej `symboling`.

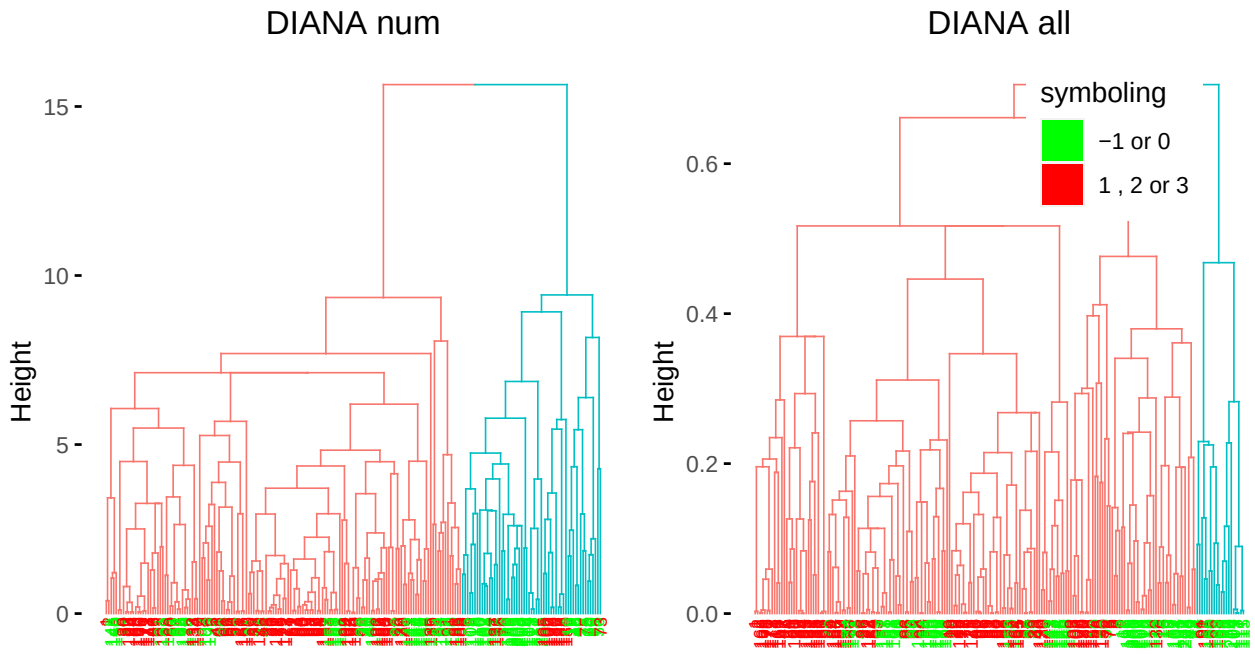



Rysunek 76: Klasteryzacja na 2 grupy za pomocą metody DIANA, a 5 grup zmiennej objaśnianej dla zmiennych normalized.losses, fuel.system, num.of.doors, length




Rysunek 77: Klasteryzacja na 2 grupy za pomocą metody DIANA, a 2 grupy zmiennej objaśnianej dla zmiennych normalized.losses, fuel.system, num.of.doors, length

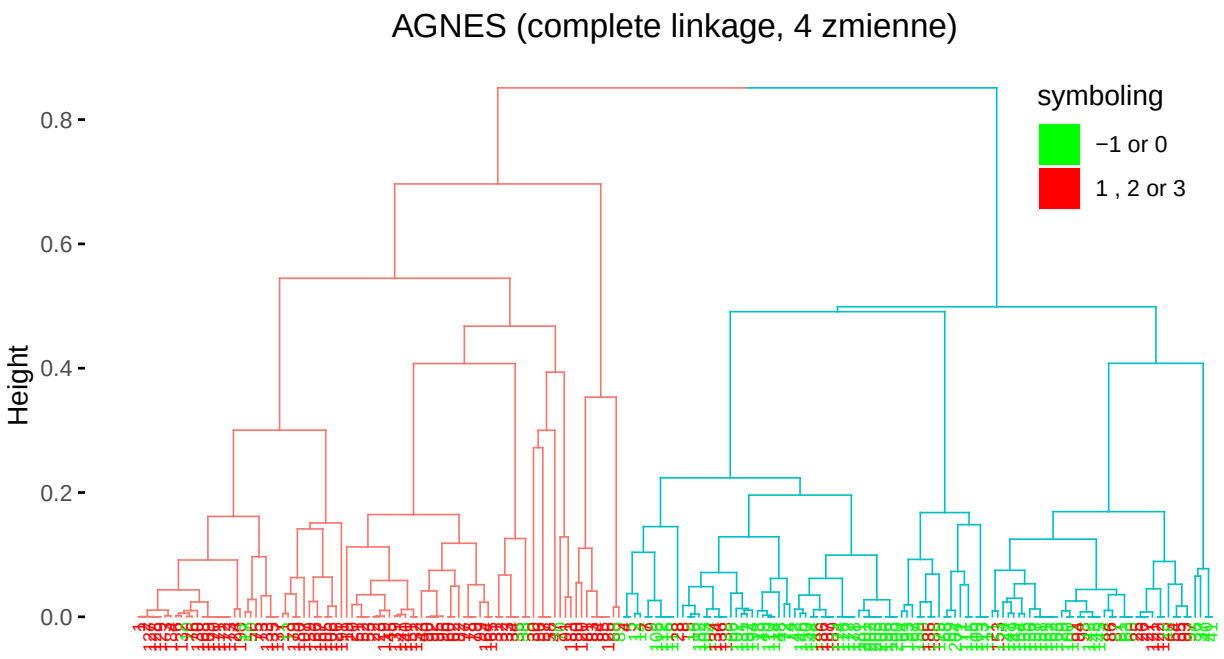
Jak możemy zaobserwować powyżej $DIANA$ użyta dla 4 wybranych zmiennych zwróciła bardzo ciekawe rezultaty. Mianowicie widzimy, że optymalną liczbą klastrów jest $k = 2$, i to własnie dla tego podziału widzimy, że 2 rozdzielone partycje dobrze oddają podział na 2 grupy zmiennej `symboling` odpowiadające za mniej oraz bardziej ryzykowne auta. Żeby potwierdzić tezę, że jest to kwestia wyboru cech, poniżej na rysunku 78 przedstawione zostały analogiczne dendrogramy dla odpowiednio zmiennych numerycznych oraz wszystkich zmiennych z zaznaczonymi etykietkami zmiennej objaśnianej `symboling` dla podziału na 2 kategorie, a także z zaznaczonym podziałem na 2 klastry w dendrogramie.




Rysunek 78: Klasteryzacja na 2 grupy za pomocą metody DIANA, a dwie grupy zmiennej objaśnianej dla zmiennych normalized.losses, fuel.system, num.of.doors, length

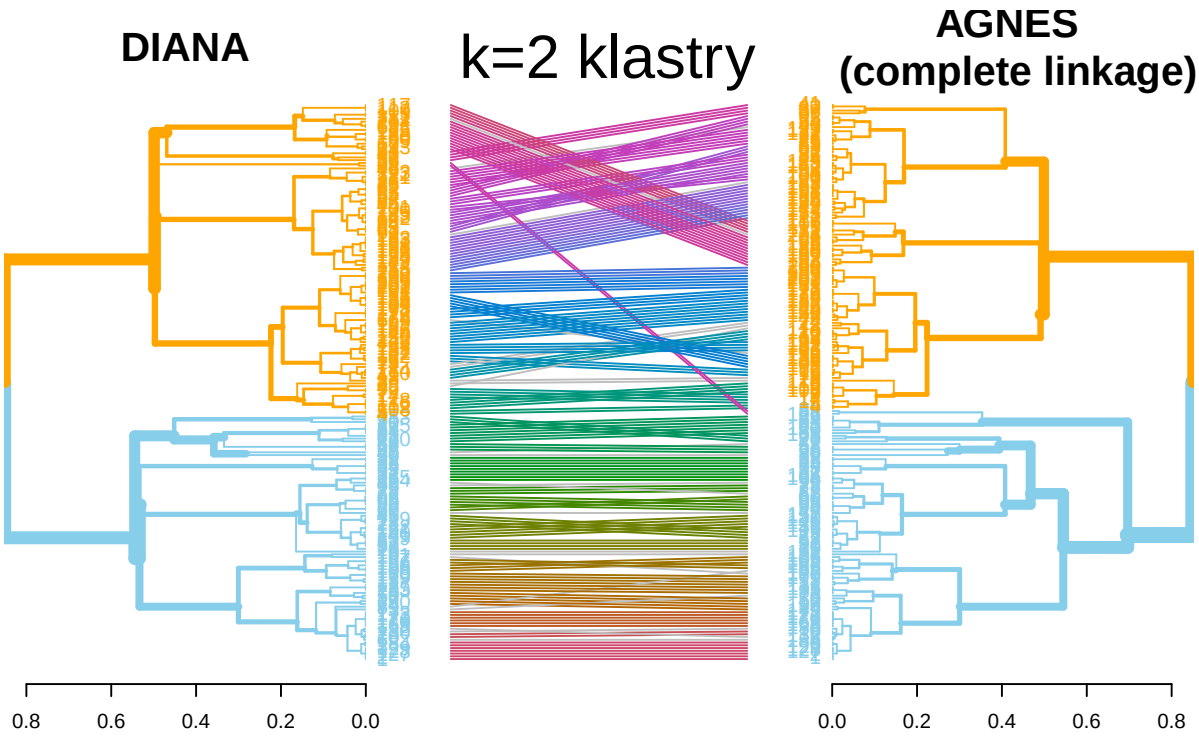
Jasno widzimy, że podzbiory zawierające większą część zmiennych nie poradziły sobie tak dobrze z separacją wybranych grup zmiennej `symboling`.

Następnie porównamy czy algorytm $AGNES$ z metodą łączenia grup complete linkage działa równie dobrze biorąc pod uwagę tylko te same 4 zmienne: `normalized.losses`, `fuel.system`, `num.of.doors`, `length`. Na rysunku 79 przedstawiony został dendrogram odpowiadający temu wyborowi cech.




Rysunek 79: Klasteryzacja na 2 grupy za pomocą metody AGNES, a dwie grupy zmiennej objaśnianej dla zmiennych normalized.losses, fuel.system, num.of.doors, length

Jak widzimy powyżej, ten dendrogram jest dużo bardziej zbliżony do analogicznego dla metody *DIANA*. Widzimy, że 2 grupy zmiennej `symboling` są dobrze uchwycone. Żeby dokładniej porównać te 2 dendrogramy, ponownie użyliśmy tanglegramu. Wyniki przedstawione zostały na rysunku 80.




Rysunek 80: Tanglegram dla metod DIANA i AGNES dla zmiennych normalized.losses, fuel.system, num.of.doors, length

Z rysunku powyżej możemy stwierdzić, że sprawdzają się nasze przypuszczenia dotyczące podobieństwa rezultatów metod *DIANA* i *AGNES*. Widzimy, że podział na 2 klastry jest bardzo podobny, prawie identyczny.

Już wcześniej wspomnieliśmy o współczynniku *AC*. Dodajmy, że dla metod dzielących istnieje analogiczny współczynnik - *DC* (divisive coefficient). Podobnie, jak w wypadku *AC*, tutaj również wartość im bliższa 1, tym bardziej istotny jest podział. Poniżej przedstawione zostały współczynniki *AC* i *DC* dla metod *AGNES* i *DIANA* odpowiednio.

```
## [1] "Współczynnik AC wynosi: 0.986"
## [1] "Współczynnik DC wynosi: 0.984"
```

I widzimy, że są one bardzo blisko 1, co pozwala nam uznać oba podziały za istotne.

Poniżej załączone zostały także fragmenty kodu z czymś na wzór tabeli kontygencji. UWAGA: etykietki klastrów przypisane są "na odwrót" tzn. poprawnie zaklsyfikowane obiekty nie leżą na głównej diagonali, a na drugiej!

```
#funkcja zwracająca etykiety klastrów
cutree(agnes_test, k=2)[1:15]

##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##  1  1  1  2  2  1  2  2  2  1  1  2  1  2  2

table(symboling_custom, cutree(agnes_test, k=2)) #klastry sa odwrotnie!!

##
## symboling_custom  1  2
```

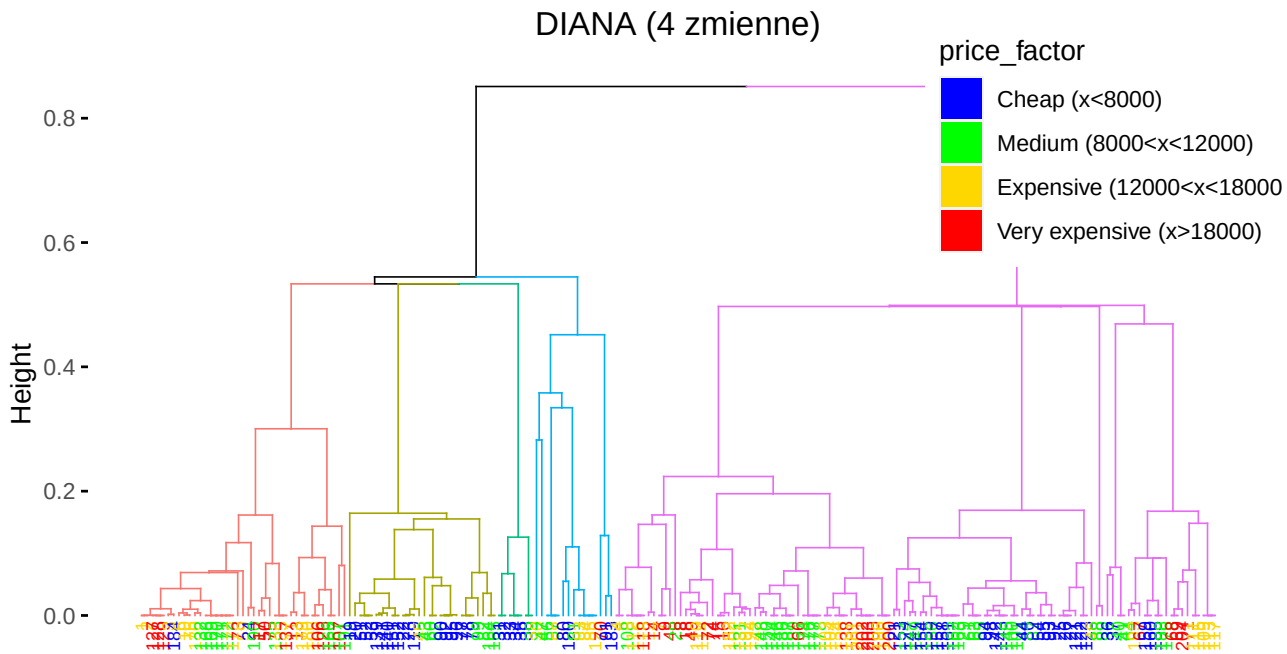
```
##       -1 or 0     8 81
##       1 , 2 or 3 83 30

#analogicznie dla Diany
cutree(diana_test, k=2)[1:15]

##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##  1  1  1  2  2  1  2  2  2  1  1  2  1  2  2

table(symboling_custom, cutree(diana_test, k=2))

##
## symboling_custom  1  2
##       -1 or 0     7 82
##       1 , 2 or 3 82 31
```

Poniżej przedstawione zostały także wyniki dokładności dla obu tych metod:

```
## [1] "Dokładność klasyfikacji dla metody AGNES: 0.81"
## [1] "Dokładność klasyfikacji dla metody DIANA: 0.81"
```

jak widzimy, obie metody zwróciły dokładnie ten sam poziom dokładności 0.81, który możemy uznać za wysoki.

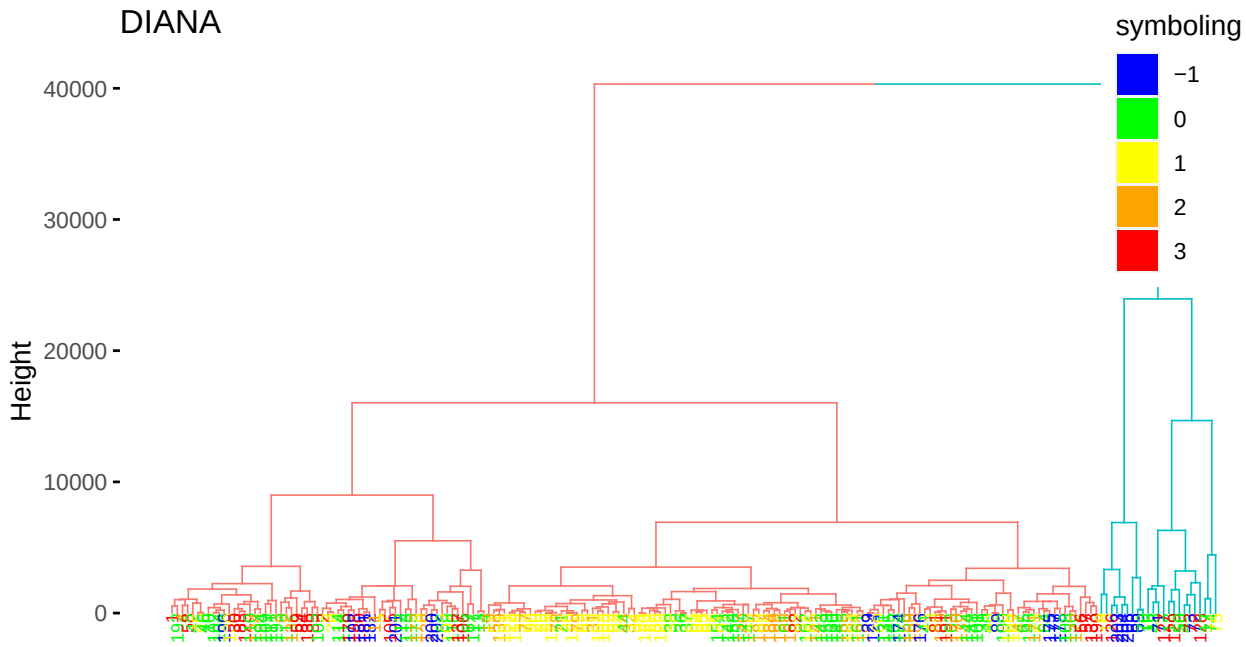
Jak pamiętamy, na początku stworzyliśmy zmienną jakościową `price`, na bazie oryginalnej. Na rysunku 81 przedstawione jest porównanie klasteryzacji na 4 grupy za pomocą metody *DIANA* z rzeczywistymi etykietkami naszej zmodyfikowanej zmiennej `price`.




Rysunek 81: Klasteryzacja na 4 grupy za pomocą metody DIANA, a dwie grupy zmiennej price

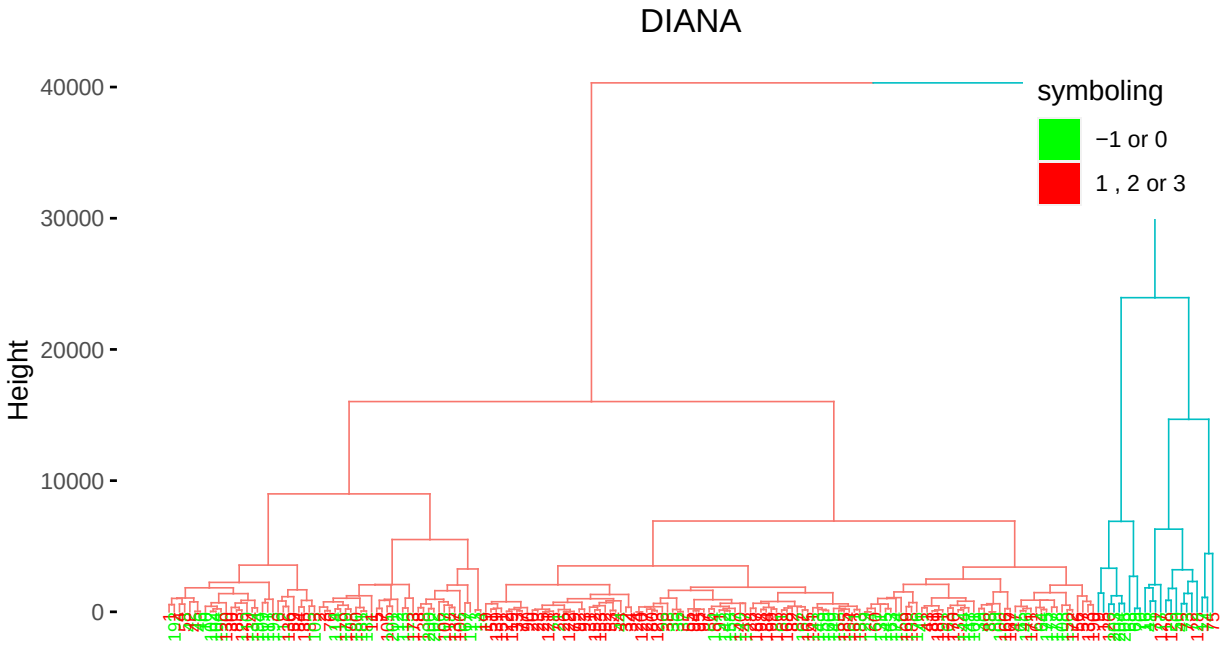
Widzimy, że wyniki naszego małego ekspetymentu nie są zbyt satysfakcjonujące. Nie możemy zauważyć, żadnego porządku i powiązania między otrzymanymi klastrami, a rzeczywistymi wartościami zmiennej `price`.

Sprawdzimy jeszcze działanie metody *DIANA* dla podzbioru 4 innych zmiennych. Przetestowanych zostało wiele podzbiorów zmiennych od 2 do 5 elementowych, mogę stwierdzić jedynie, że dla żadnego podzbioru nie uzyskaliśmy dokładności większej niż 0.81, a przeważnie kończyło się na gorszych rezultatach. Dodatkowo wartym podkreślenia jest fakt, że przy testowaniu ręcznie "istotności" zmiennych, najważniejszą wydawała się `num.of.doors`, od której wystepowania lub jego braku w rozważanym podzbiorze cech zależało najwięcej. Poniżej na rysunkach 82, **??** umieszczone zostały przykładowe dendrogramy dla zmiennych `price`, `height`, `curb.weight`, `wheel.base` i `peak.rpm`.




Rysunek 82: Klasteryzacja na 2 grupy za pomocą metody DIANA, a dwie grupy zmiennej objaśnianej dla zmiennych price, height, curb.weight, wheel.base i peak.rpm

```
## [1] "Dokładność dla zmiennej symboling przyjmującej tylko dwie wartości: 0.59"
```




Rysunek 83: Klasteryzacja na 2 grupy za pomocą metody DIANA, a dwie grupy zmiennej objaśnianej dla zmiennych price, height, curb.weight, wheel.base i peak.rpm

Z powyżych rysunków widzimy jednoznacznie, że dla tego podzbioru wyniki nie są już tak zadowalające.

# 5 Redukcja wymiaru

Redukcja wymiaru jest bardzo ważną częścią procesu data mining. Umożliwia ona m.in wizualizacje wyników na wykresach (dzięki sprowadzeniu do danych dwuwymiarowych), eliminacje nadmiarowej informacji czy identyfikacje struktury zależności w danych. Metody redukcji wymiaru mogą należeć zarówno do uczenia nienadzorowanego jak i do uczenia nadzorowanego w zależności od konkretnej metody. Jednakże dwie wybrane przez nas metody: $PCA$ i $MDS$ są przykładami uczenia nienadzorowanego.

## 5.1 PCA - analiza składowych głównych

Pierwszą metodą redukcji wymiaru wybraną przez nas jest $PCA$. Streszczając, poszukuje ona zbioru złożonej z mniejszej liczby zmiennych, dzięki któremu uda się nam zachować najważniejsze własności wyjściowych danych. Metoda ta działa bardzo dobrze zwłaszcza, gdy w analizowanych danych występują silnie skorelowane zmiene. Dodać trzeba, że $PCA$ działa tylko na zmiennych ilościowych. Metoda ta działa w ten sposób, że wybieramy kilka pierwszych składowych głównych, które wyjasniają odpowiednio dużą część całkowitej zmienności danych. Więcej o $PCA$ tutaj: [23].

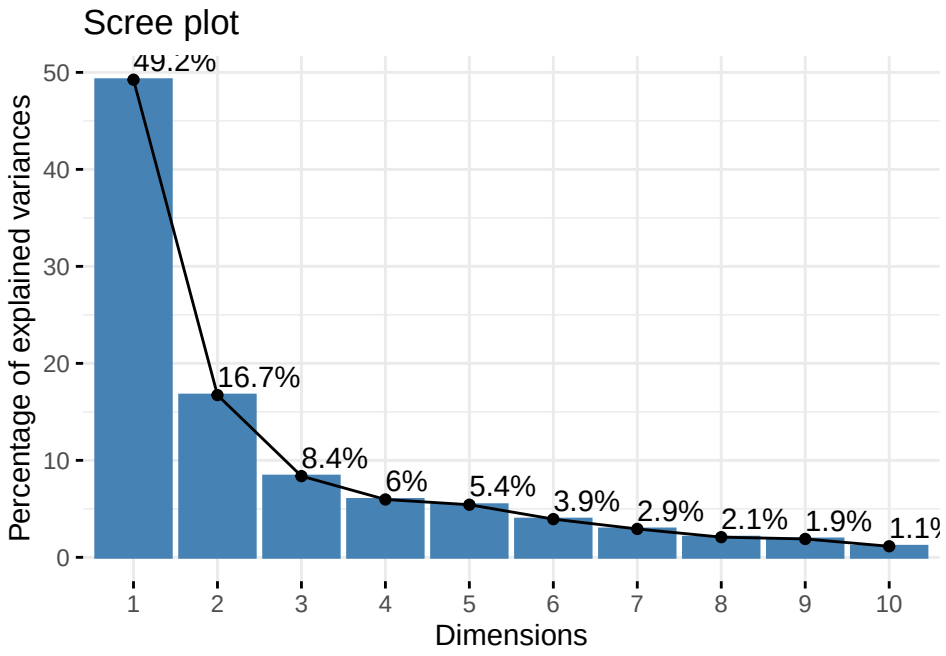
Do implementacji metody $PCA$ w **R** będziemy używać funkcji `PCA` [24] z pakietu **FactoMiner**. Poniżej przedstawiony został zbiór elementów zwracany przez tą funkcję.

```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 202 individuals, described by 15 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"

## starting httpd help server ... done
```

Poniżej przedstawiony został fragment kodu z wartościami własnymi macierzy kowariancji oraz wariancje składowych głównych.

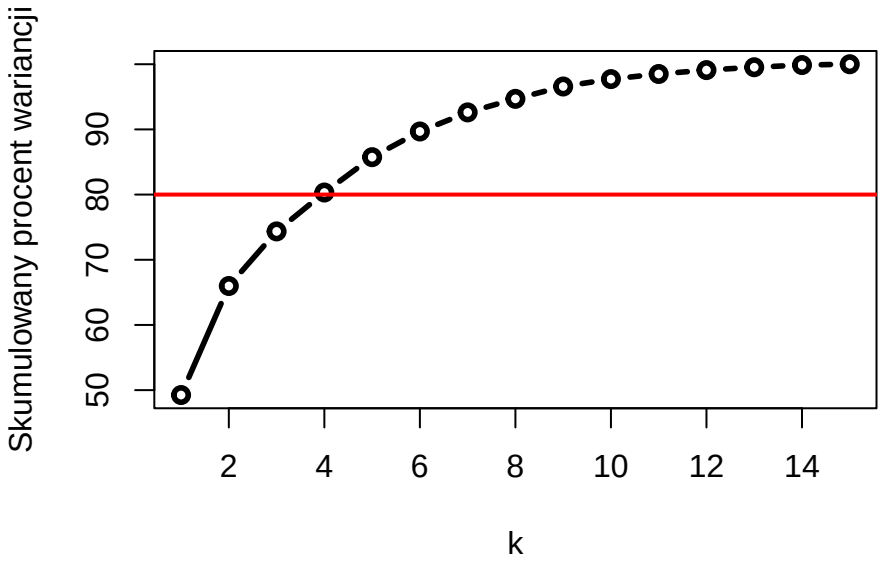
```
##        eigenvalue variance.percent cumulative.variance.percent
## Dim.1  7.38694501       49.2463001                    49.24630
## Dim.2  2.50935638       16.7290425                    65.97534
## Dim.3  1.25624854        8.3749903                    74.35033
## Dim.4  0.89483536        5.9655691                    80.31590
## Dim.5  0.81373679        5.4249119                    85.74081
## Dim.6  0.59111185        3.9407456                    89.68156
## Dim.7  0.43896423        2.9264282                    92.60799
## Dim.8  0.31206672        2.0804448                    94.68843
## Dim.9  0.28411186        1.8940790                    96.58251
## Dim.10 0.17076900        1.1384600                    97.72097
## Dim.11 0.11708338        0.7805559                    98.50153
## Dim.12 0.08972921        0.5981948                    99.09972
## Dim.13 0.06540077        0.4360051                    99.53573
## Dim.14 0.05072099        0.3381399                    99.87387
## Dim.15 0.01891992        0.1261328                   100.00000
```

Poniżej na rysunku 84 zilustrowane zostały wariancje składowych głównych.




Rysunek 84: Wariancje składowych głównych w metodzie PCA

Jednakże lepszym sposobem analizy jest porównanie skumulowanej wariancji pierwszych $k$ składowych głównych. Poniżej na rysunku 85 został przedstawiony własnie taki wykres, z dodaną linią przy wartości 80 procent.

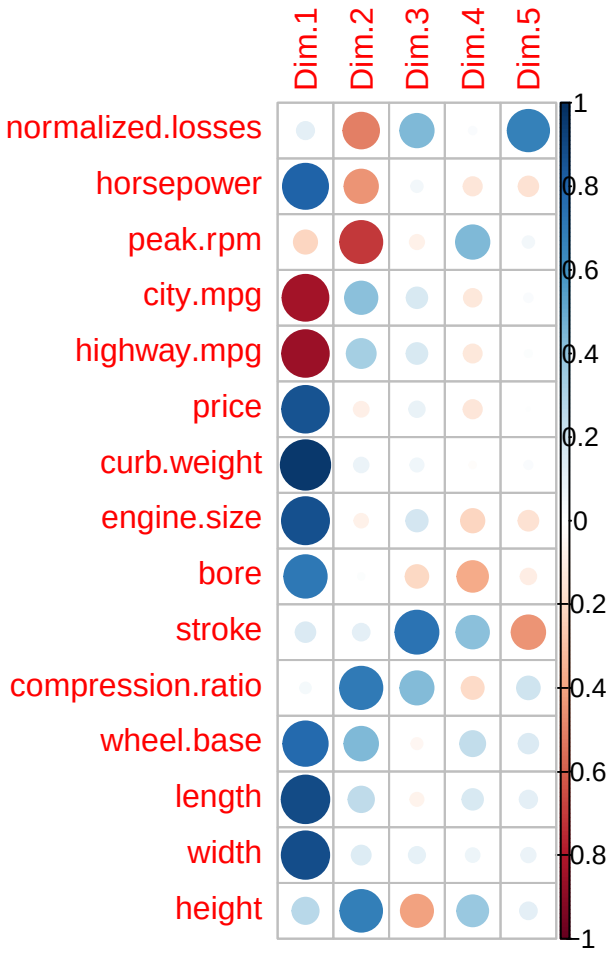



Rysunek 85: Skumulowana wariancja pierwszych k składowych głównych

Widzimy, że pierwsze 4 składowe główne odpowiadają za około 80 procent całej wariancji. Poniżej przedstawione są ładunki poszczególnych zmiennych dla pierwszych 5 składowych

głównych. Jeszcze niżej, na rysunku 86 przedstawione zostały korelacje zmienne z poszczególnymi składowymi głównymi.

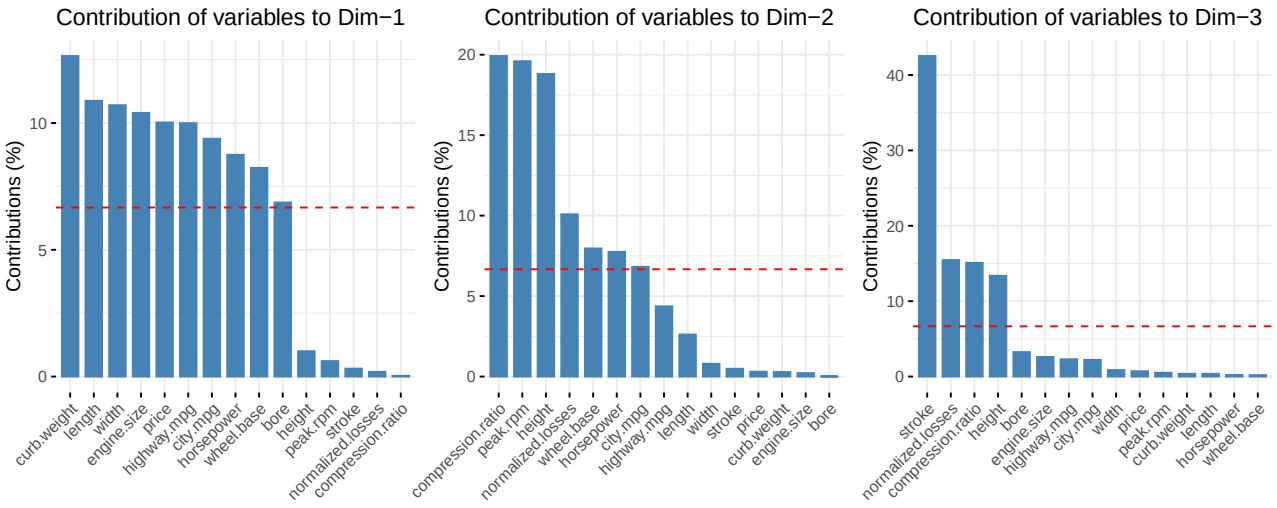
```
##                        Dim.1       Dim.2       Dim.3       Dim.4       Dim.5
## normalized.losses  0.04299907 -0.31752013  0.39308333  0.02326830  0.750876807
## horsepower         0.29573530 -0.27841980  0.04641015 -0.13810572 -0.167054028
## peak.rpm          -0.07820718 -0.44266490 -0.07100704  0.46571636  0.057994277
## city.mpg          -0.30627911  0.26108699  0.14884793 -0.12846194  0.031987432
## highway.mpg       -0.31615699  0.20889895  0.15145551 -0.13500816  0.021247414
## price              0.31657394 -0.05489313  0.08419581 -0.13868626  0.003855721
## curb.weight        0.35557675  0.05316525  0.06062066 -0.01760471  0.029231152
## engine.size        0.32248581 -0.04626442  0.16134351 -0.22733288 -0.171440920
## bore               0.26202792  0.01022530 -0.18021122 -0.39205984 -0.106300039
## stroke             0.05587999  0.06965573  0.65210368  0.44284361 -0.491948674
## compression.ratio  0.01653042  0.44632328  0.38841240 -0.20273186  0.226085850
## wheel.base         0.28689080  0.28215058 -0.04333975  0.26158000  0.167102255
## length             0.32980179  0.16160441 -0.06045406  0.17627930  0.132572920
## width              0.32715682  0.08929685  0.09276061  0.07870337  0.090117093
## height             0.10000317  0.43366573 -0.36562623  0.40115539  0.123003376
```




Rysunek 86: Skumulowana wariancja pierwszych k składowych głównych

Widzimy, że aż 10 na 15 zmiennych numerycznych jest silnie skorelowanych już z pierwsza składową główną.

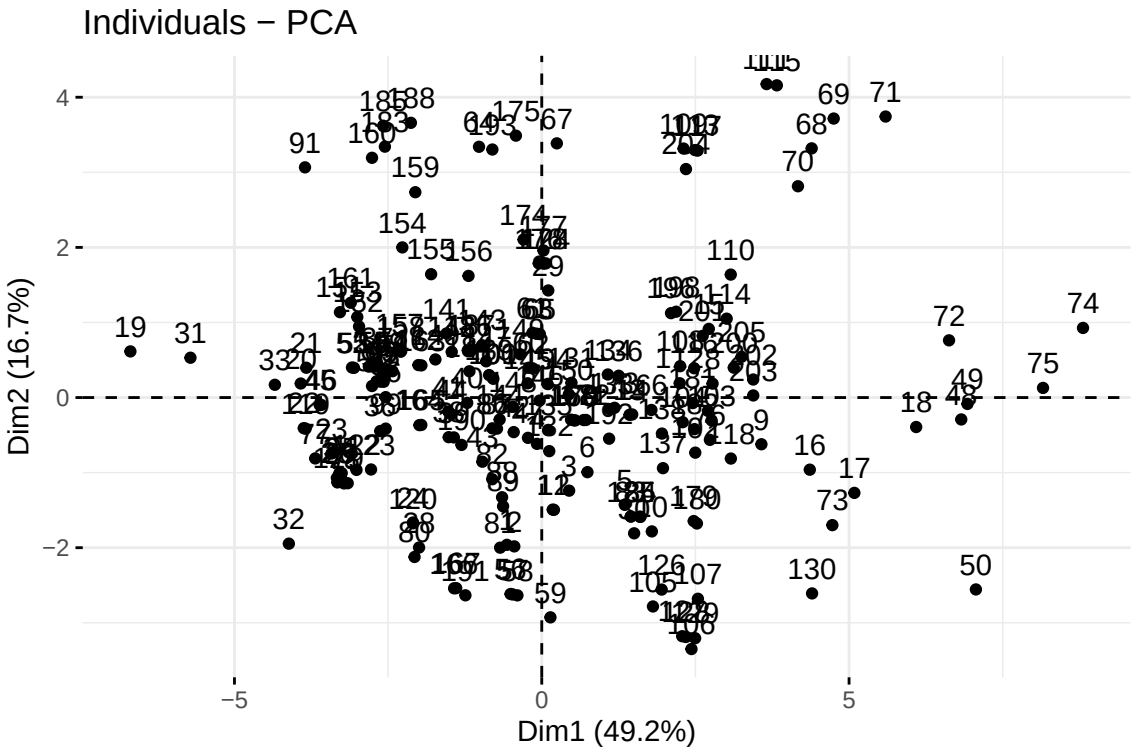
Poniżej na rysunku 87 zobrazowane zostały wkłady poszczególnych zmiennych w pierwsze 3 składowe główne.




Rysunek 87: Skumulowana wariancja pierwszych k składowych głównych

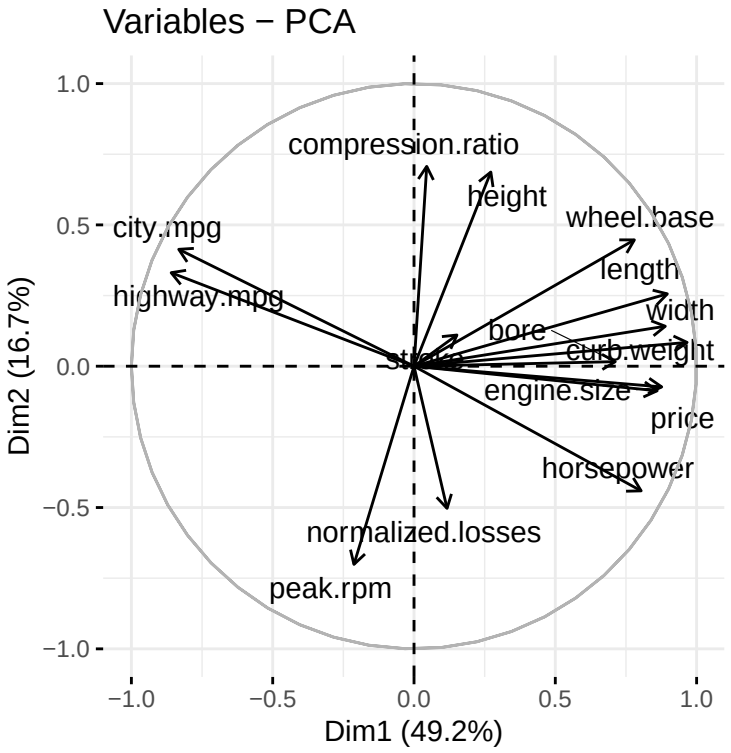
Na wykresie powyżej widzimy, że największy wkład w pierwszą składową główną ma zmienna `curb.weight`. Oprócz niej największy wkład mają `length`, `width` i `engine.size`, z kolei zdecydowanie najmniejszy wpływ mają zmienne `height`, `peak.rpm`, `stroke`, `normalized.losses` oraz `compression.ratio`. Natomiast w przypadku drugiej składowej głównej, największy wpływ mają `compression.ratio`, `peak.rpm` i `height`, czyli zmienne, które miały znikomy wpływ na pierwszą składową. W trzeciej składowej głównej zdecydowanie największą rolę odgrywa zmienna `stroke`, która ma bardzo mały wkład w dwie pierwsze składowe główne.

Z kolei poniżej na rysunku 88, zamieszczony został wykres punktowy, który przedstawia rozrzut poszczególnych obiektów w przestrzeni dwóch pierwszych składowych głównych.



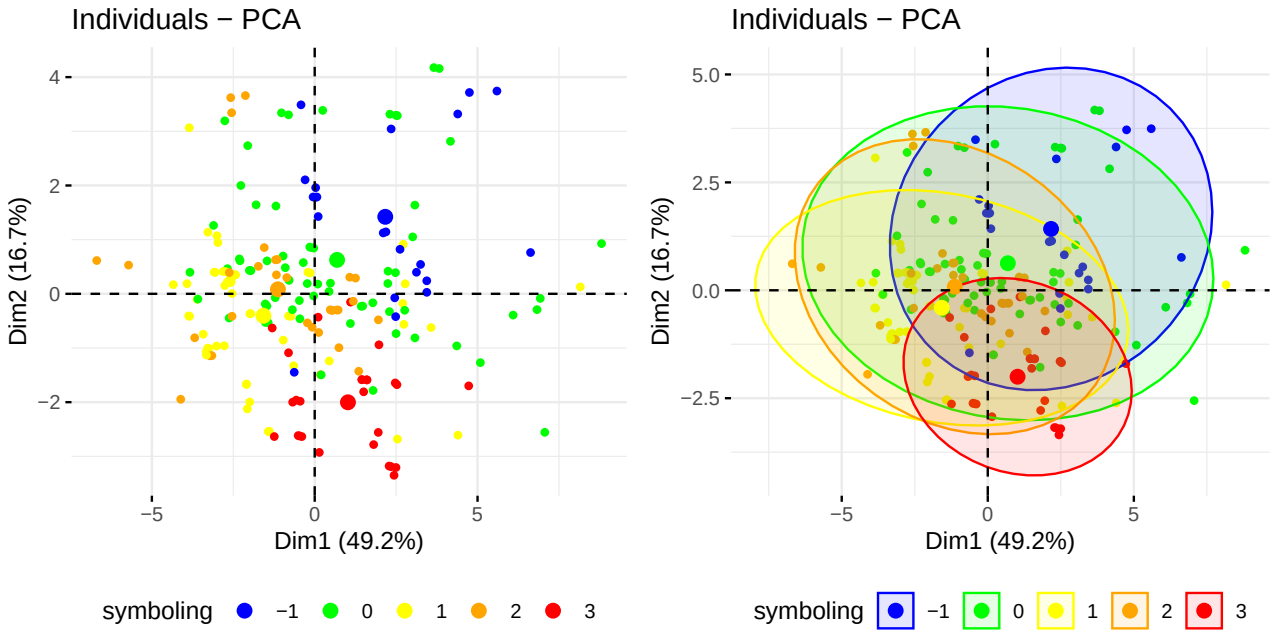

Rysunek 88: Wykres rozrzutu w przestrzeni dwóch pierwszych składowych głównych

Poniżej na rysunku 89, zamieszczony został wykres, który przedstawia rozmieszczenie poszczególnych zmiennych w przestrzeni dwóch pierwszych składowych głównych.

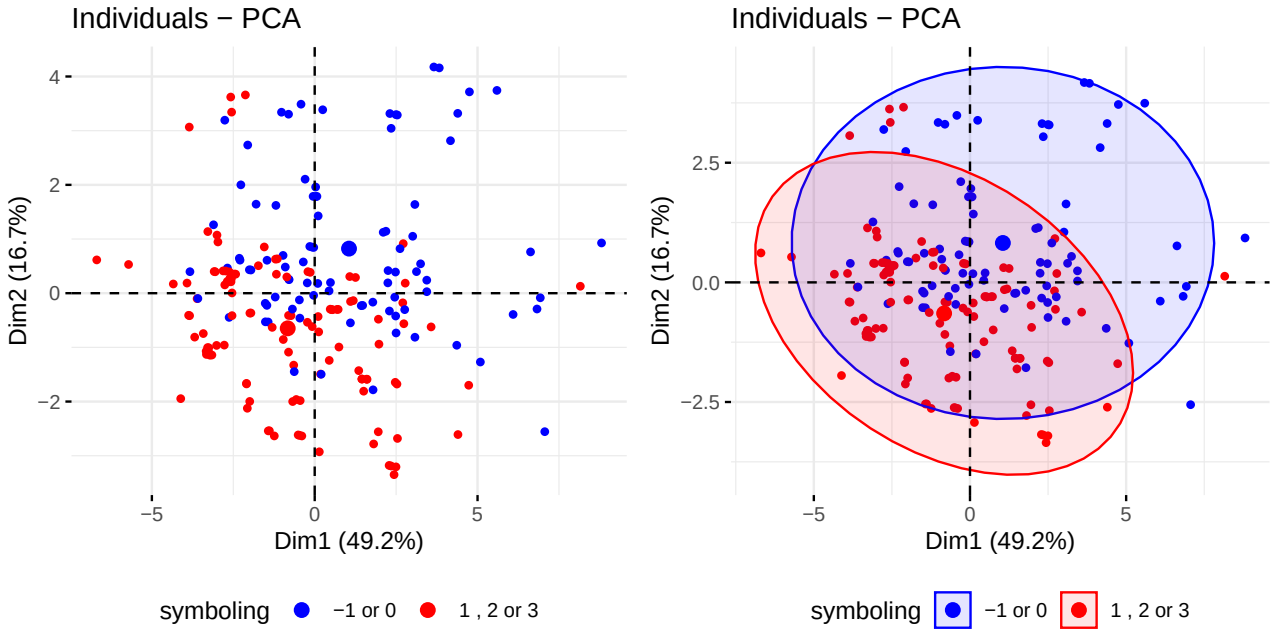



Rysunek 89: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych

Teraz przejdziemy do porównania rezultatów otrzymanych metodą *PCA* (konketnie pierwszych 2 składowych) z rzeczywistymi wartościami zmiennych jakościowych. Na początek, na rysunkach 90, 91 przedstawione zostało porównanie ze zmienną `symboling`.



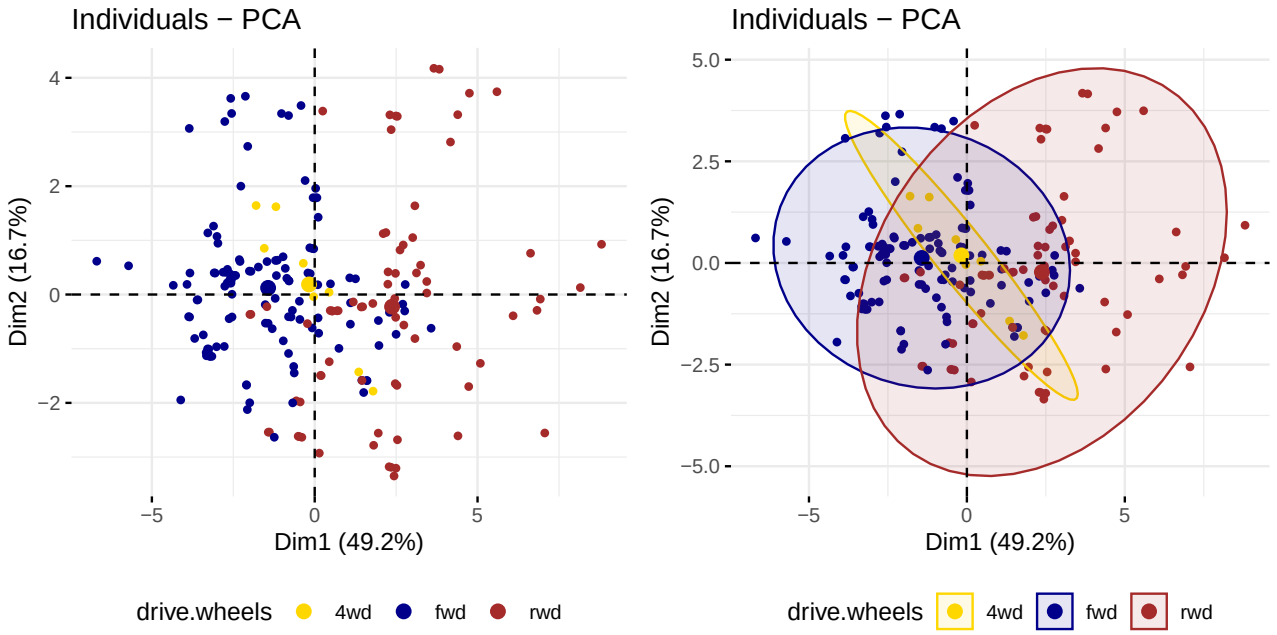

Rysunek 90: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych




Rysunek 91: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych

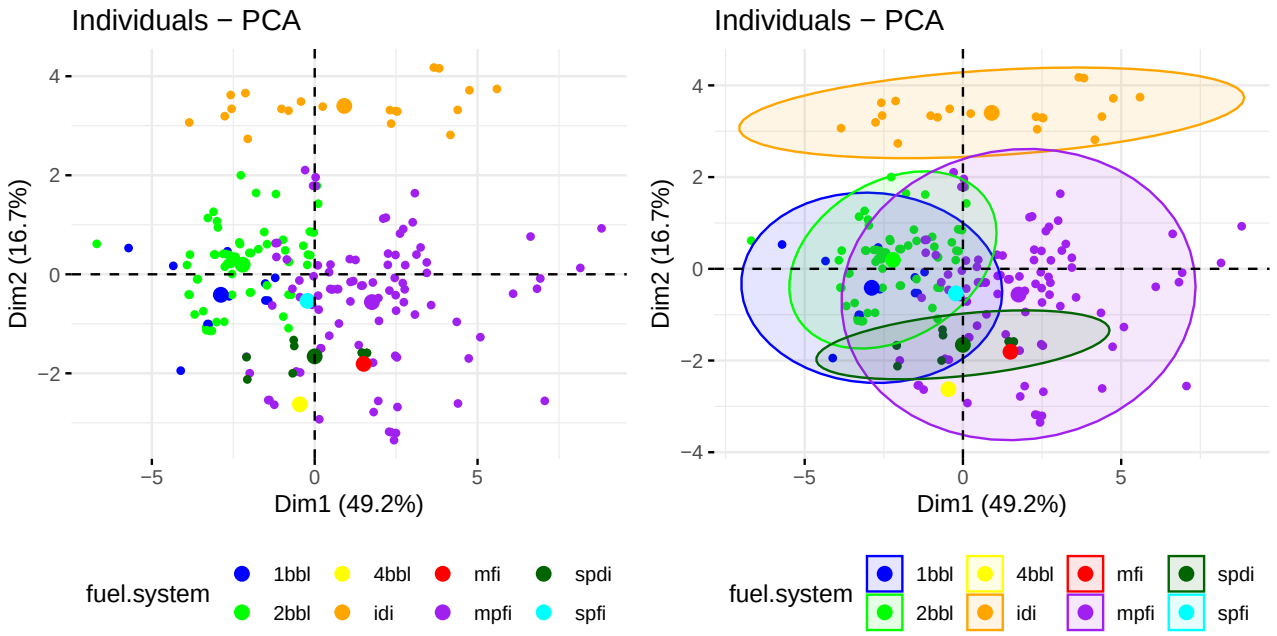
Mimo że możemy dostrzec pewne wzorce na powyższych rysunkach, to możemy stwierdzić, że kategorie zmiennej `symboling` nie zostały prawidłowo wychwycone.

Poniżej, na rysunkach 92, 93, 94 przedstawione zostały analogiczne wizualizacje dla innych zmiennych jakościowych: `drive.wheels`, `fuel.system` i `num.of.doors`.

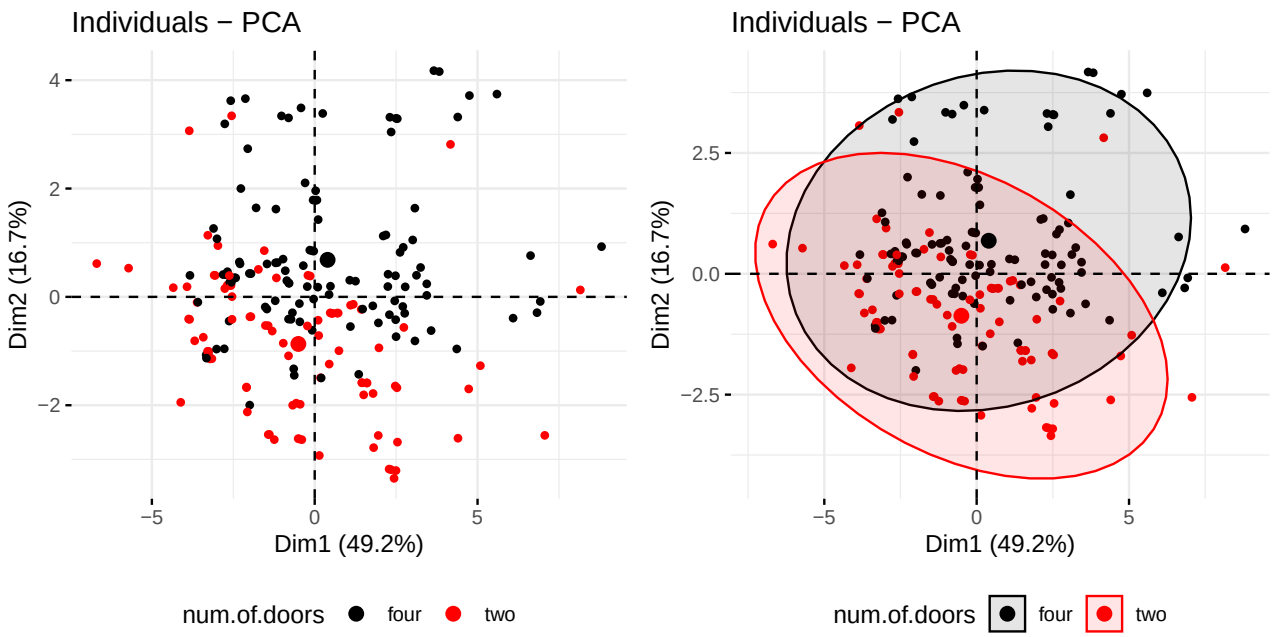



Rysunek 92: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych

```
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
```



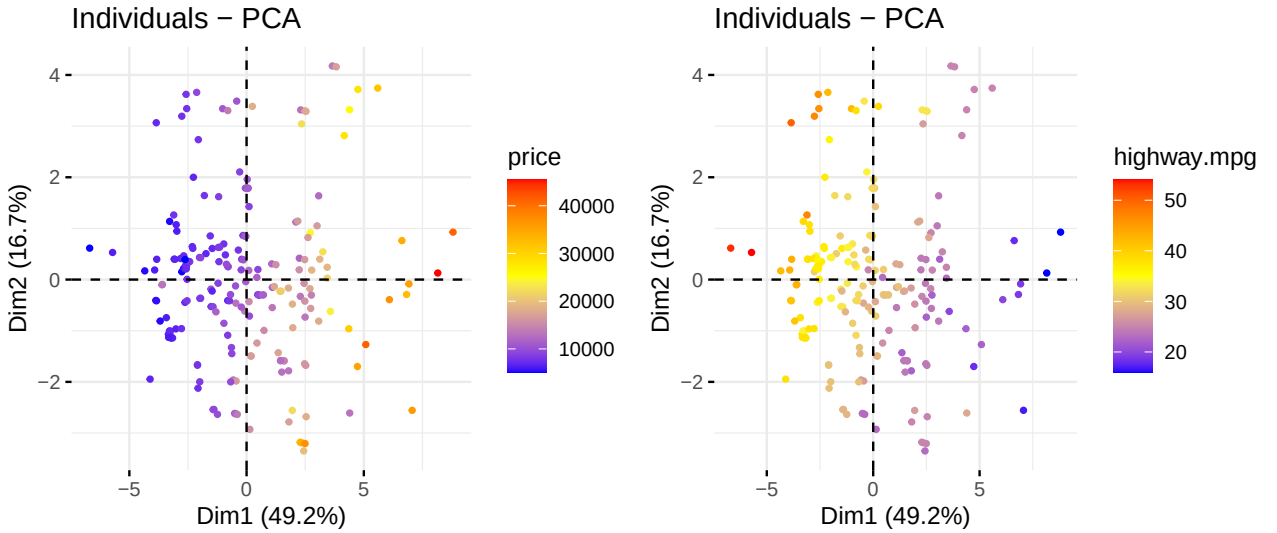

Rysunek 93: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych




Rysunek 94: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych

Z rysunków powyżej możemy wywnioskować, że żadna ze zmiennych jakościowych nie została odseparowana w bardzo dobry sposób. Możemy wychwycić pewne wzorce, jak np. dla `fuel.system`="idi", ale to wszystko.

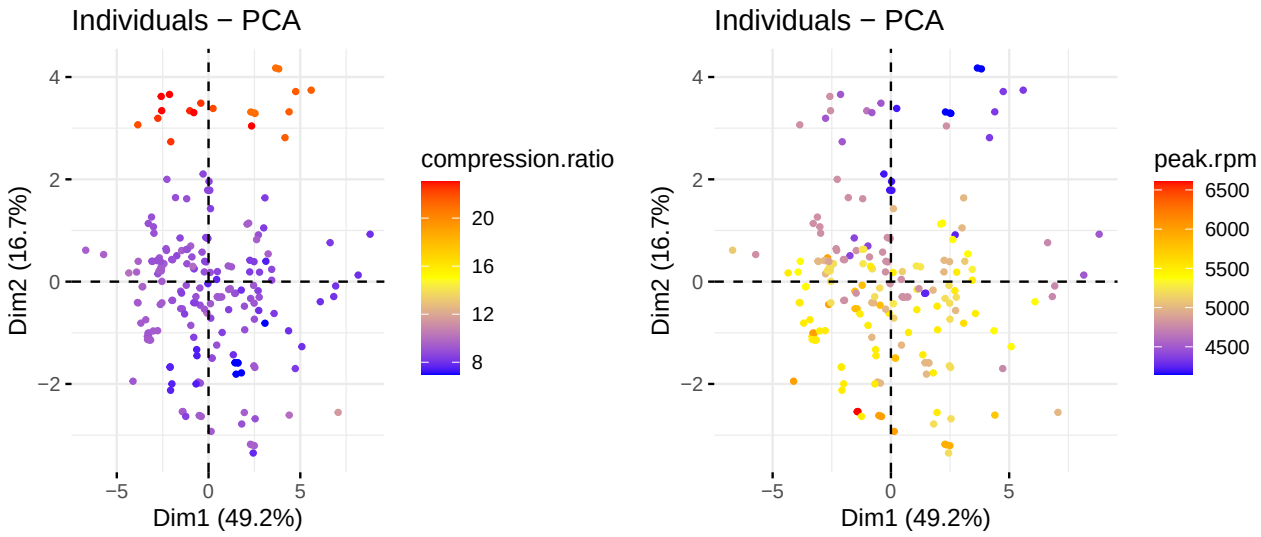
Ponadto, na rysunku 95 przedstawione zostały 2 wykresy punktowe porównujące rozrzut zmiennych ilościowych `price` i `highway.mpg` tzn. wszystkie obiekty zaznaczone są kolorami zgodnymi z odpowiednimi wartościami tych zmiennych.




Rysunek 95: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych

Możemy zaobserwować, że zmienna `price` skorelowana jest dodatnio z pierwszą składową główną, a zmienna `highway.mpg` skorelowana jest ujemnie z tą samą składową.

Poniżej na rysunku 96 analogiczne wykresy dla zmiennych `compression.ratio` i `peak.rpm`



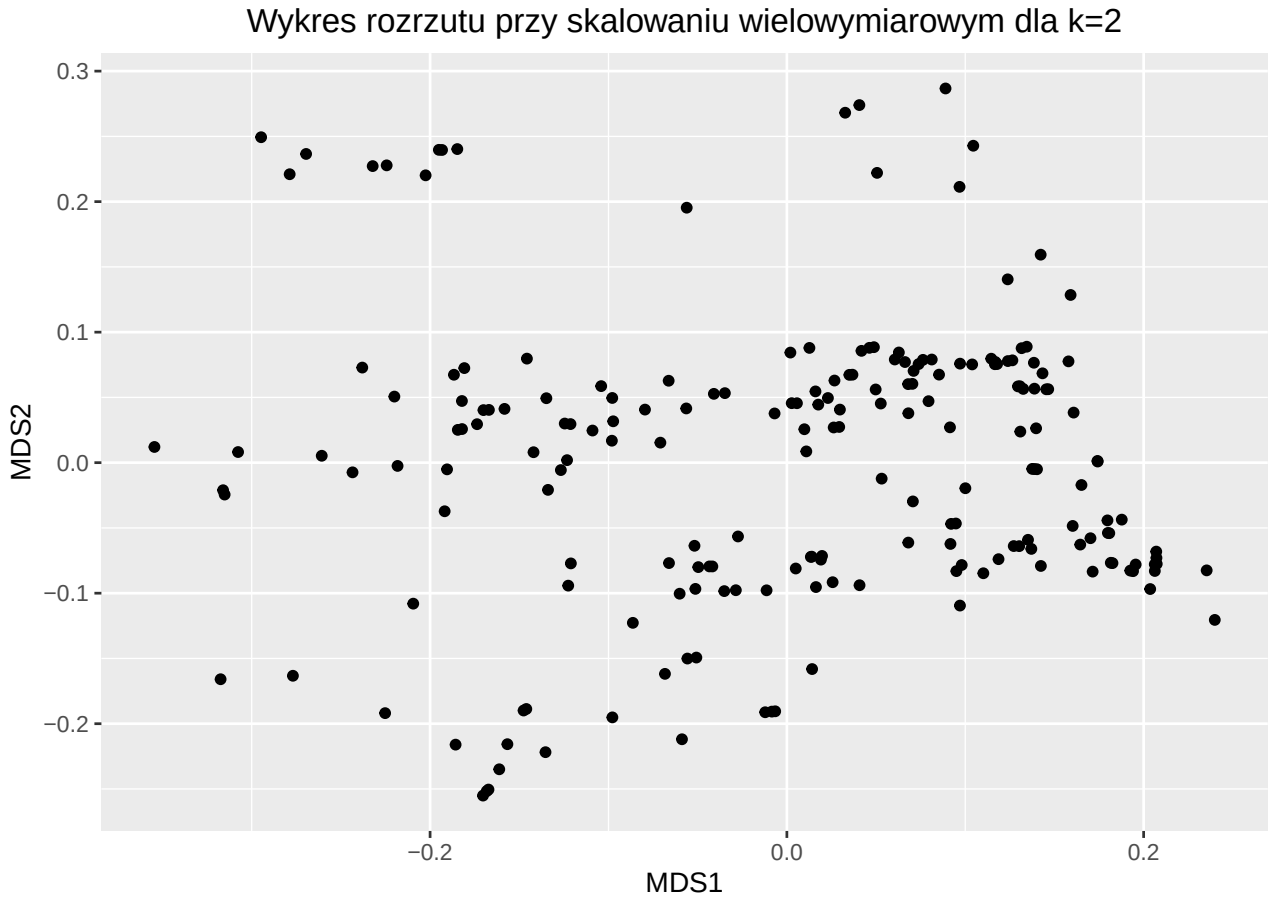

Rysunek 96: wpływ poszczególnych zmiennych na wartości dwóch pierwszych składowych głównych

Widzimy powyżej, że zmienna `compression.ratio` skorelowana jest dodatnio z drugą składową głowną, a zmienna `peak.rpm` skorelowana ujemnie. Wszystkie te wnioski są potwierdzeniem danych, które dostaliśmy na rysunku 89. Uwaga: przy zmiennych ilościowych, które porównywaliśmy na 2 powyższych wykresach, należy pamiętać, że nie usuwaliśmy ich z podzbioru wybranych cech, co mogło mieć duże znaczenie na rezultaty.

## 5.2 MDS - skalowanie wielowymiarowe

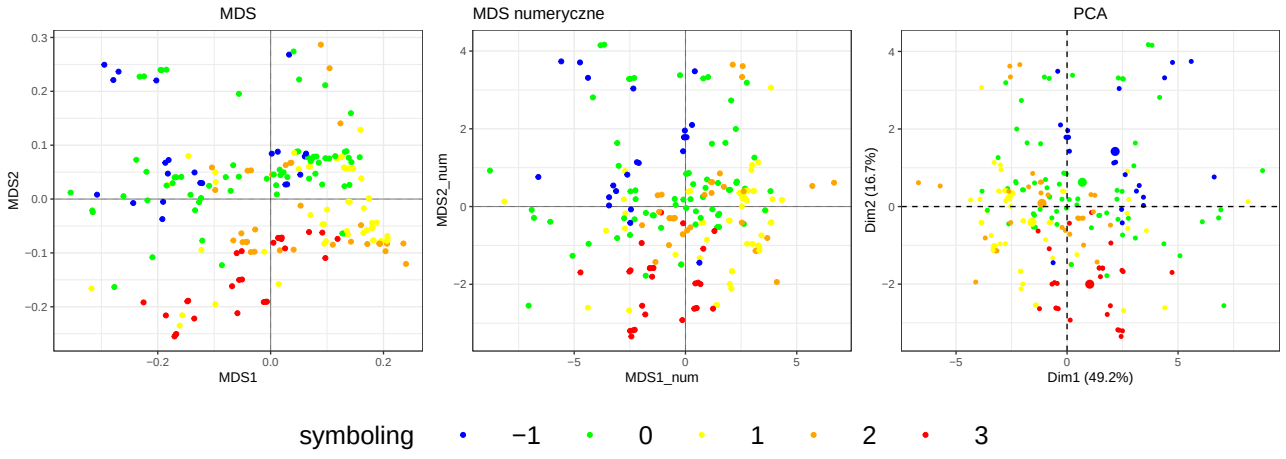
Drugą metodą redukcji wymiaru użytą przez nas będzie $MDS$ czyli skalowanie wielowymiarowe. Metoda ta polega na odtworzeniu odległości (lub ogólniej odmienności) między obiektami w nowej przestrzeni o mniejszym wymiarze. Główną zaletą metody $MDS$ jest właśnie fakt, że na wejściu możemy jej dostarczyć dowolną macierz odmienności, czyli działa ona także na zmiennych jakościowych. Więcej o metodzie $MDS$ tutaj: [25].

Na rysunku 97 przedstawiony został wykres rozrzutu przy skalowaniu wielowymiarowym do wymiaru $k = 2$ dla wszystkich zmiennych. Do implementacji $MDS$ w **R** użyliśmy funkcji `cmdscale` [26] z pakietu **stats**.




Rysunek 97: Wykres rozrzutu przy skalowaniu wielowymiarowym dla k=2

Poniżej na rysunku 98 przedstawione zostało porównanie metod: $MDS$ dla wszystkich zmiennych, $MDS$ dla zmiennych numerycznych i metody $PCA$.

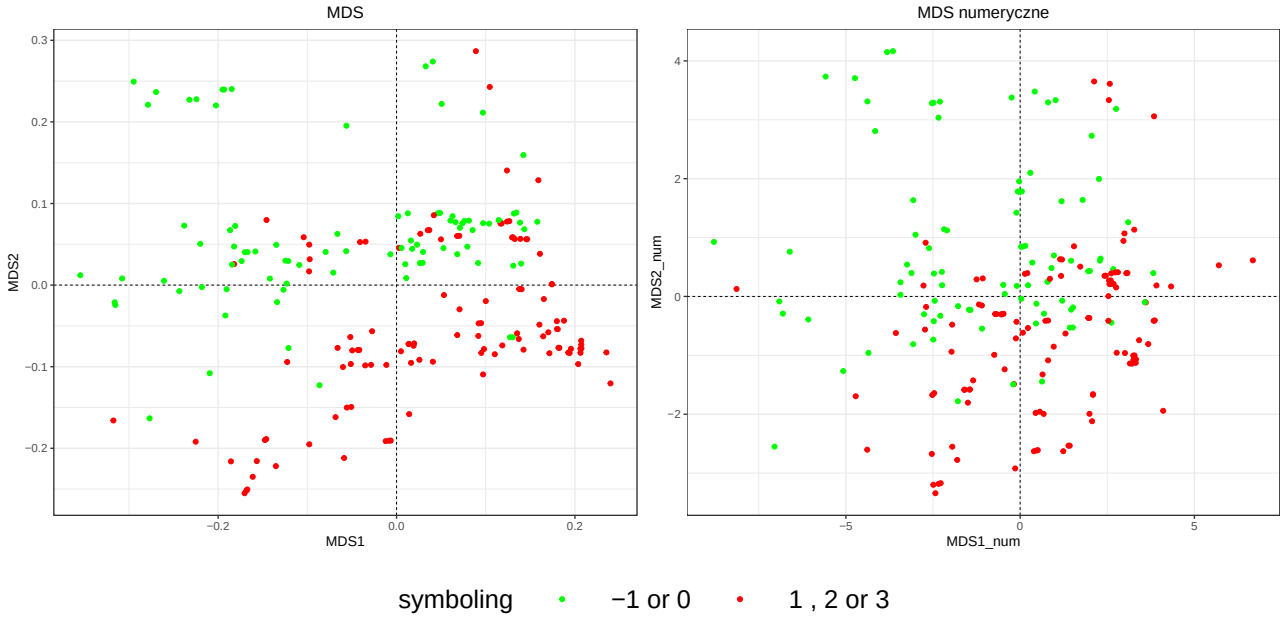



Rysunek 98: Porównanie metod MDS i PCA

Warto zauważyć znany fakt, że dla wybranych zmiennych numerycznych metoda MDS z wymiarem $k = 2$ odpowiada 2 pierwszym składowym głównym z metody PCA. (Wykresy te są symetryczne względem osi $y$). Dla pozostałych zmiennych poniżej przedstawionie zostanie już

tylko metoda $MDS$ na wszystkich zmiennych, oraz metoda $MDS$ dla zmiennych numerycznych, która odpowiada metodzie $PCA$.

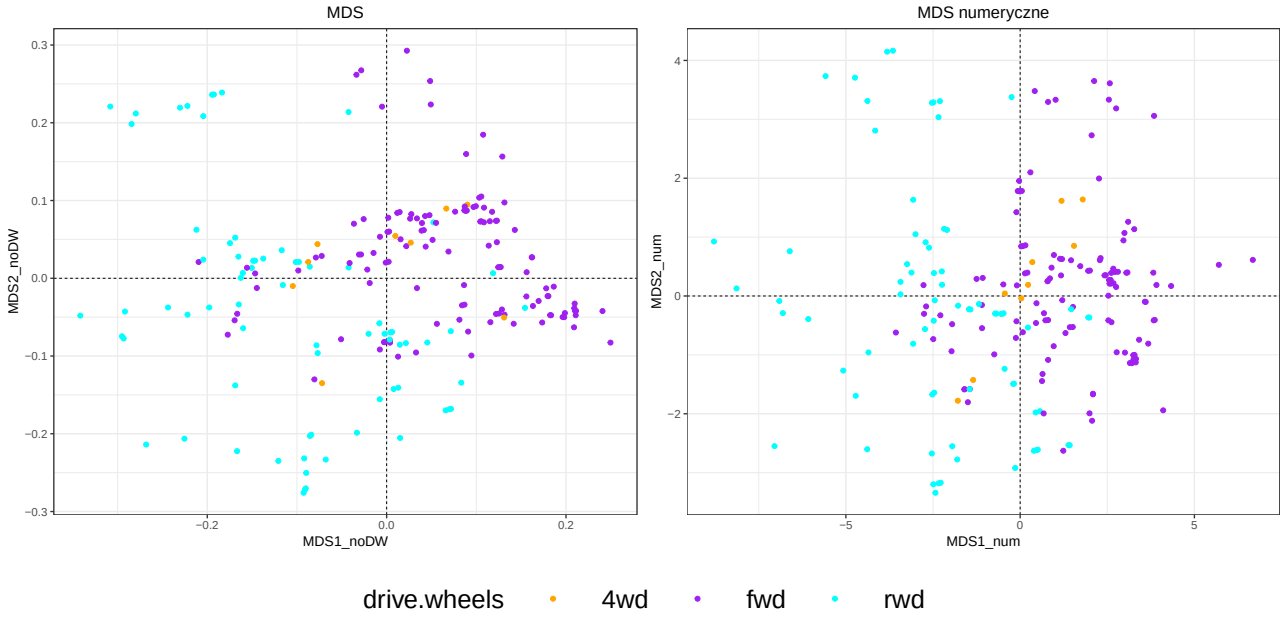
Poniżej na rysunku 99 zostały przedstawione te same wykresy co powyżej, ale dodatkowo z zaznaczonymi faktycznymi kategoriami zmiennej `symboling` w wersji z jej dwoma etykietkami.



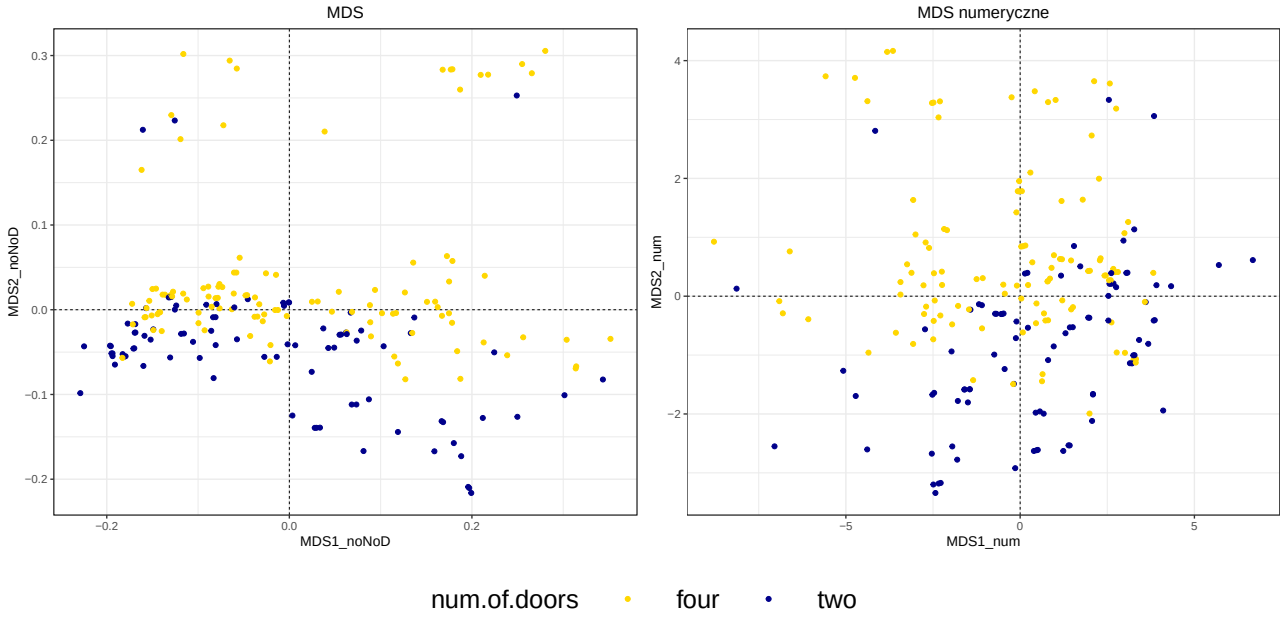

Rysunek 99: Porównanie metod MDS z rzeczywistymi wartościami zmiennej symboling

NA podstawie rysunku powyżej możemy stwierdzić, że widać pewien wzorzec rozmieszczenia poszczególnych kategorii zmiennej `symboling`, jednakże z drugiej strony nie możemy mówić też o jednoznacznej bezbłędnej separacji obu obszarów.

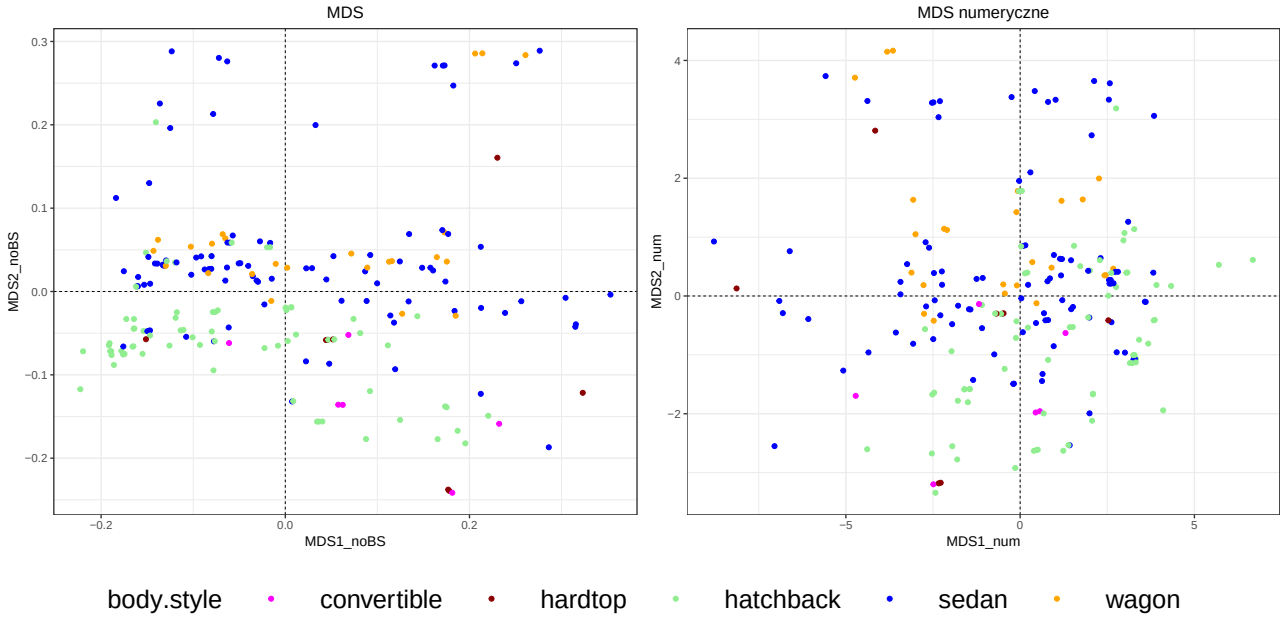
Poniżej na rysunkach 100, 101, 102, 103 i 104 przedstawione zostały analogiczne wykresy dla innych zmiennych jakościowych. Warto podkreślić, że dla każdej z nich stworzona została odpowiednia macierz odmienności nieuwzględniająca tej konkretnej zmiennej.



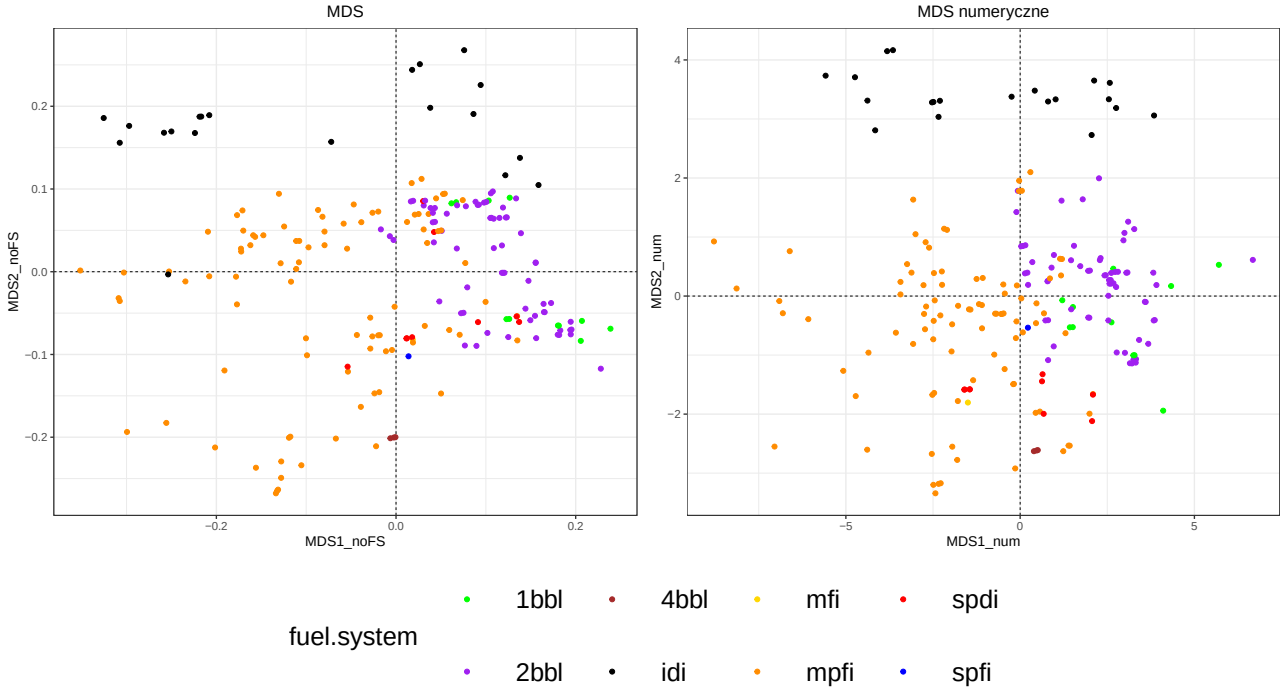

Rysunek 100: Porównanie metod MDS z rzeczywistymi wartościami zmiennej drive.wheels



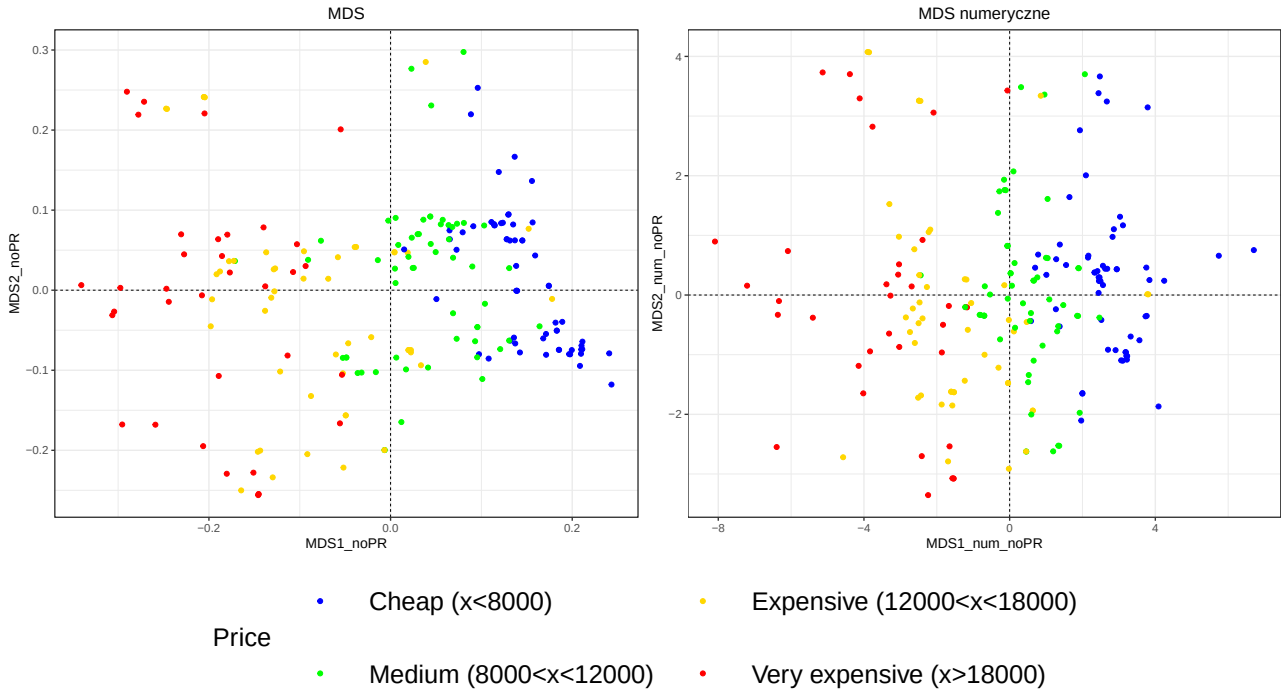

Rysunek 101: Porównanie metod MDS z rzeczywistymi wartościami zmiennej num.of.doors




Rysunek 102: Porównanie metod MDS z rzeczywistymi wartościami zmiennej body.style




Rysunek 103: Porównanie metod MDS z rzeczywistymi wartościami zmiennej fuel.system




Rysunek 104: Porównanie metod MDS z rzeczywistymi wartościami zmiennej price (typu factor)

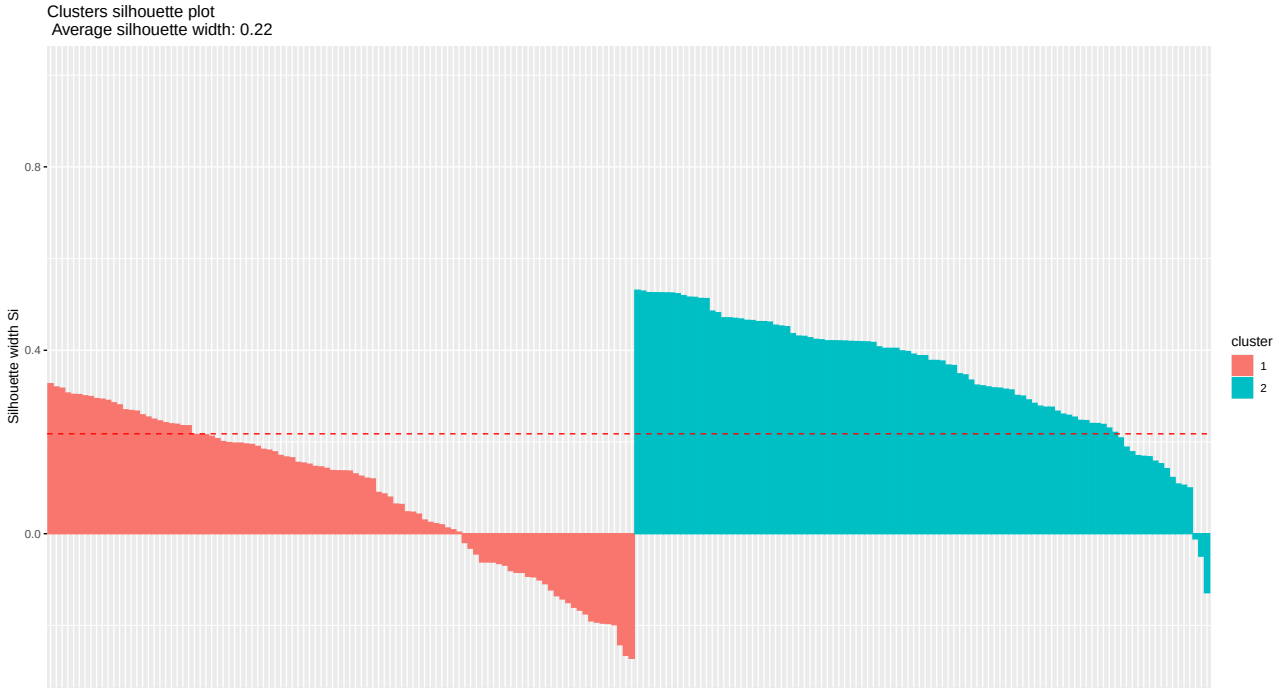
Na powyższych wykresach, widzimy, że dla wszystkich wybranych zmiennych widać pewne wzorce na wykresach rozrzutu, odnośnie odseparowania ich poszczeólnych kategorii, najgorzej prawdopodobnie dla zmiennej `body.style`. Innym wnioskiem może być to, że dla wszystkich zmiennych metoda $MDS$ użyta na bazie wszystkich zmiennych zwraca lepsze rezultaty, potencjalne partycje są bardziej widoczne niż w przypadku tylko zmiennych ilościowych.

## 5.3 Zastosowanie metod redukcji wymiaru przy klasteryzacji

Oczywiście, ważnym zastosowaniem metod redukcji wymiaru, jest po prostu użycie ich w celu klasyfikacji czy klasteryzacji.

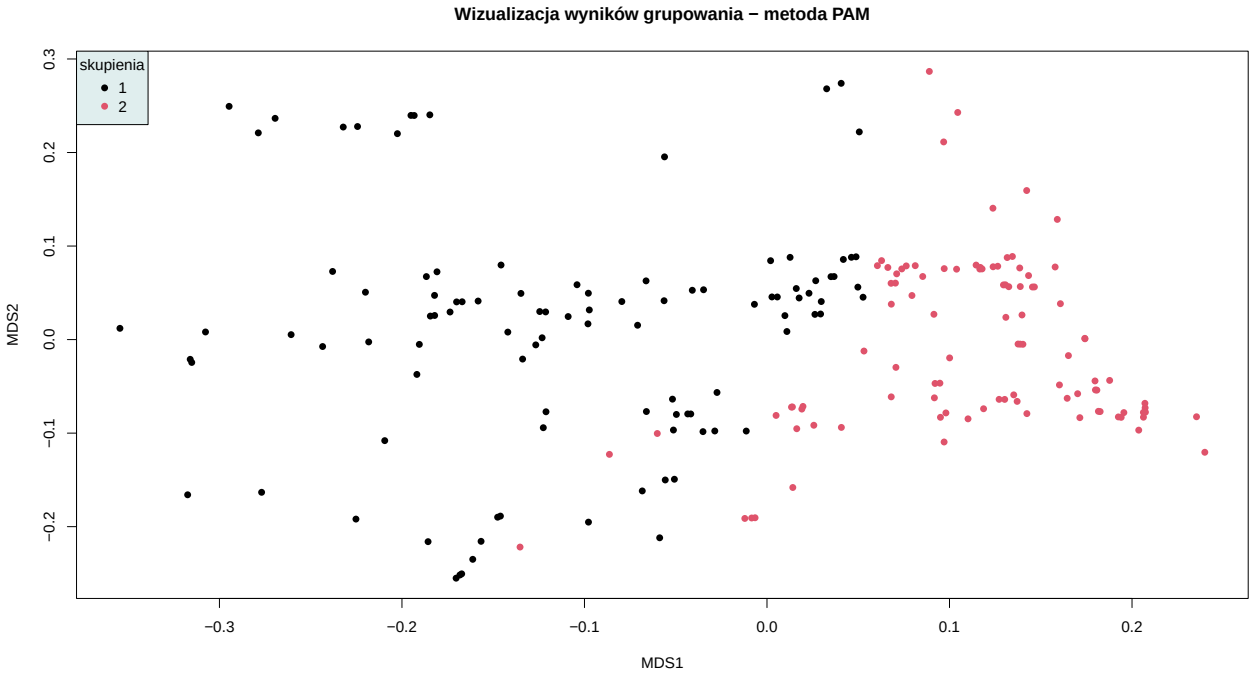
Poniżej na rysunku 105 przedstawione został wykres Silhouette'a dla metody PAM użytej na oryginalnych danych.

```
##   cluster size ave.sil.width
## 1       1  102          0.09
## 2       2  100          0.35
```




Rysunek 105: PAM na bazie MDS z k=2

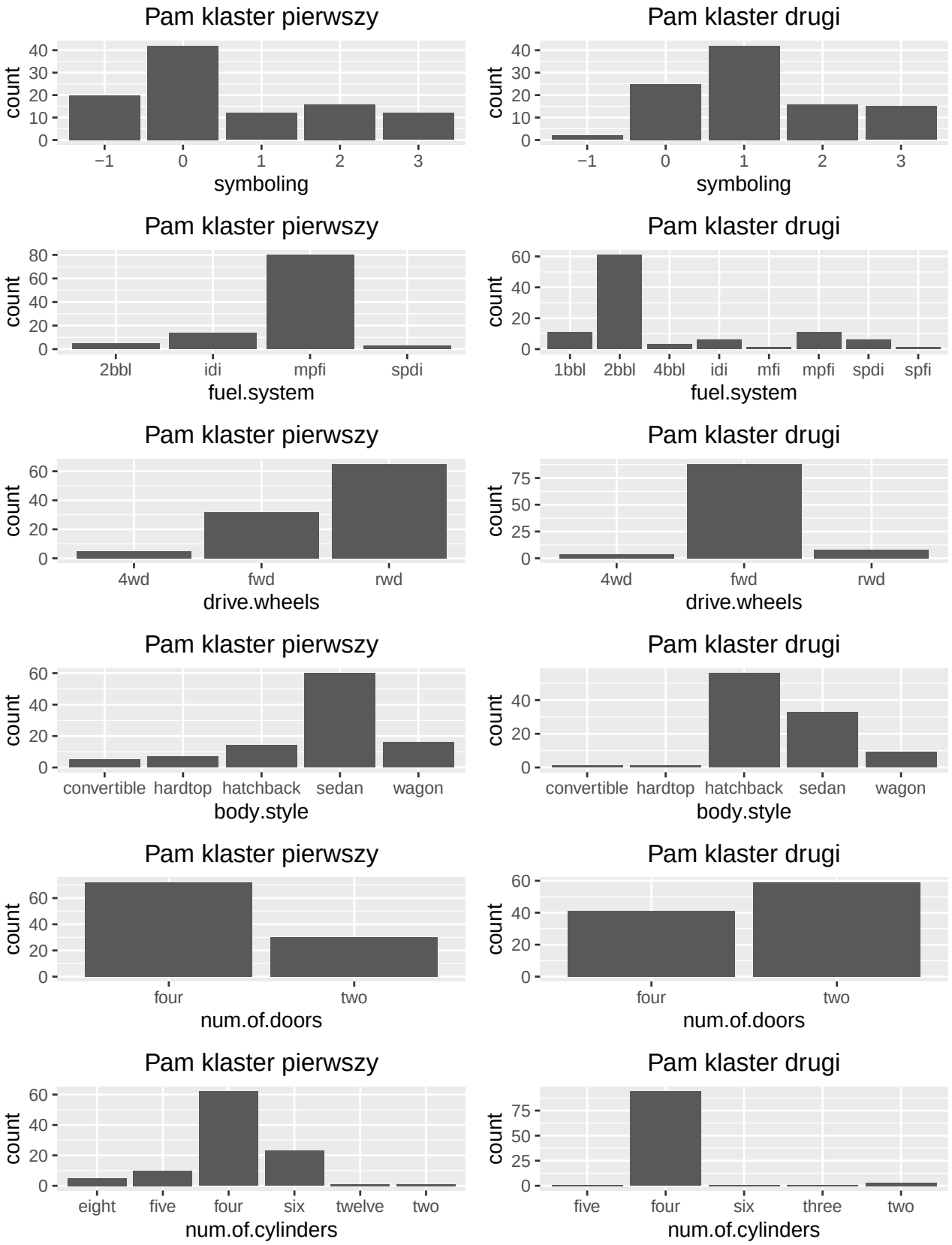
Następnie, na rysunku 106 przedstawiona została Wizualizacja wyników grupowania metodą PAM po użyciu metody $MDS$ dla wymiaru $k = 2$.




Rysunek 106: Wizualizacja wyników grupowania - metoda PAM po użyciu metody MDS

Powyżej widzimy dwa dobrze odseparowane skupienia, z małymi wyjątkami. Jednakże to jeszcze nam nic nie mówi, postaramy się zbadać charakterystykę obu tych skupień pod kątem innych zmiennych.

Na początku, na rysunku 107 przedstawione zostały wykresy słupkowe dla zmiennych jakościowych, pogrupowane ze względu na oba klastry uzyskane przy użyciu $PAM$.

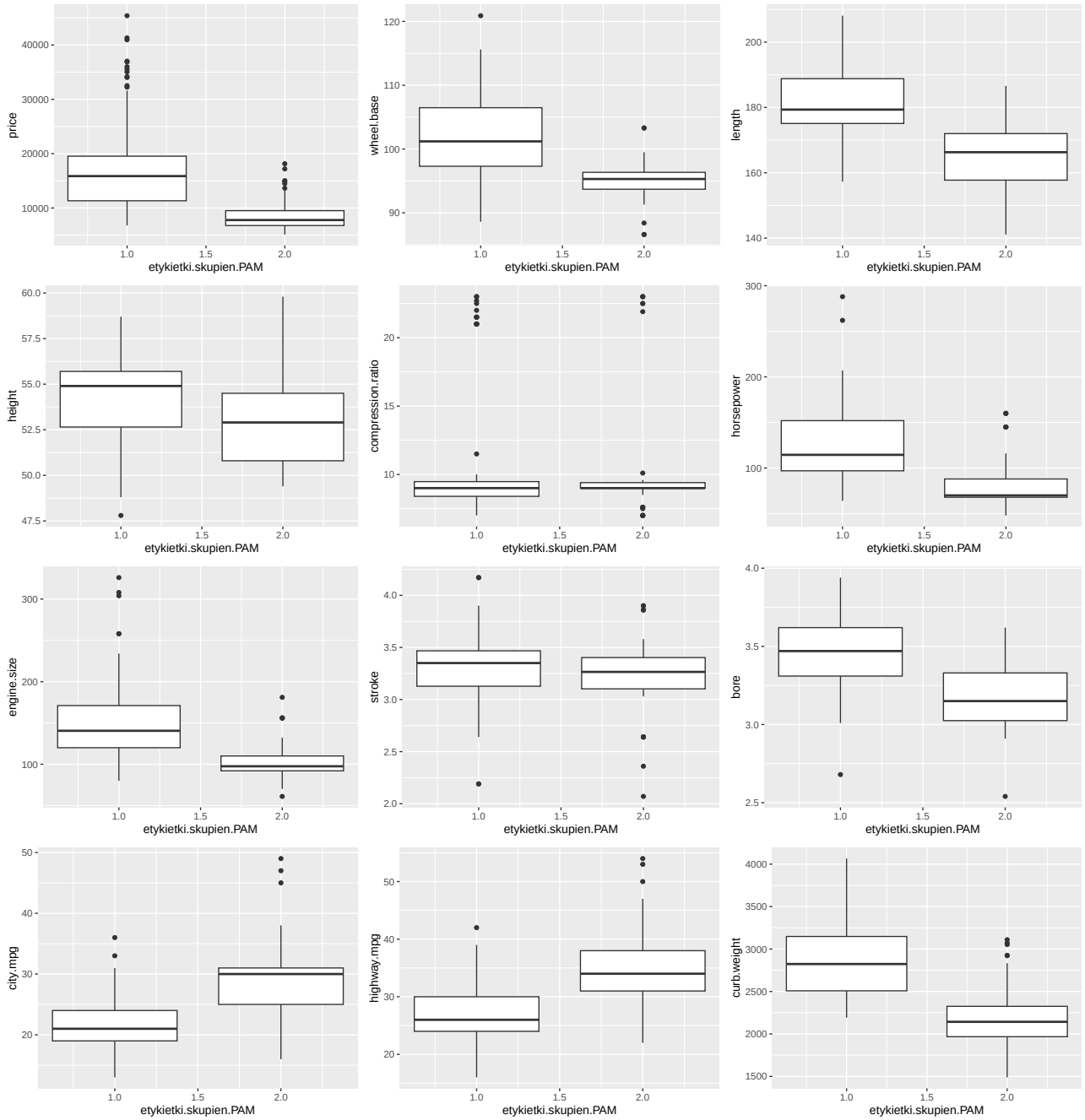



Rysunek 107: Porównanie występowania zmiennych jakościowych w dwóch klastrach po użyciu metod PAM i MDS

Jak możemy zaobserwować powyżej, rozkłady występowań poszczególnych zmiennych jako-

ściowych różnią się w zależności od klastrów. Najbardziej widać to dla zmiennych `fuel.system`, `drive.wheels`, `body.style` i `num.of.doors`.

Poniżej na rysunku 108 przedstawione zostały wykresy pudełkowe dla zmiennych ilościowych pogrupowane ze względu na klastry uzyskane metodą $PAM$.



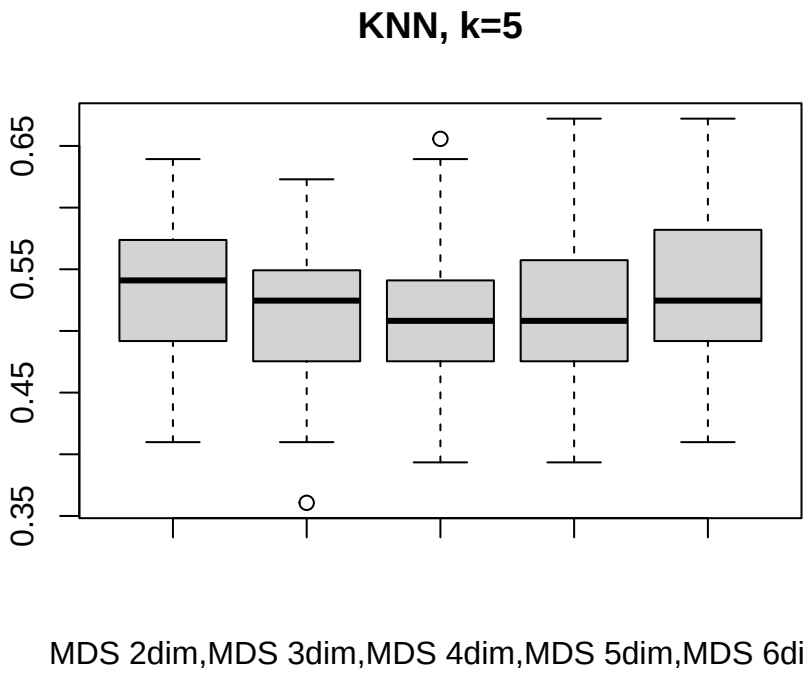

Rysunek 108: Porównanie zmiennych ilościowych dla dwóch klastrów po użyciu metod PAM i MDS

Na rysunkach powyżej widzimy wyraźne różnice dla wszystkich zmiennych ilościowych oprócz `stroke` i `compression.ratio` ze względu na klastry uzyskane metodą $PAM$.

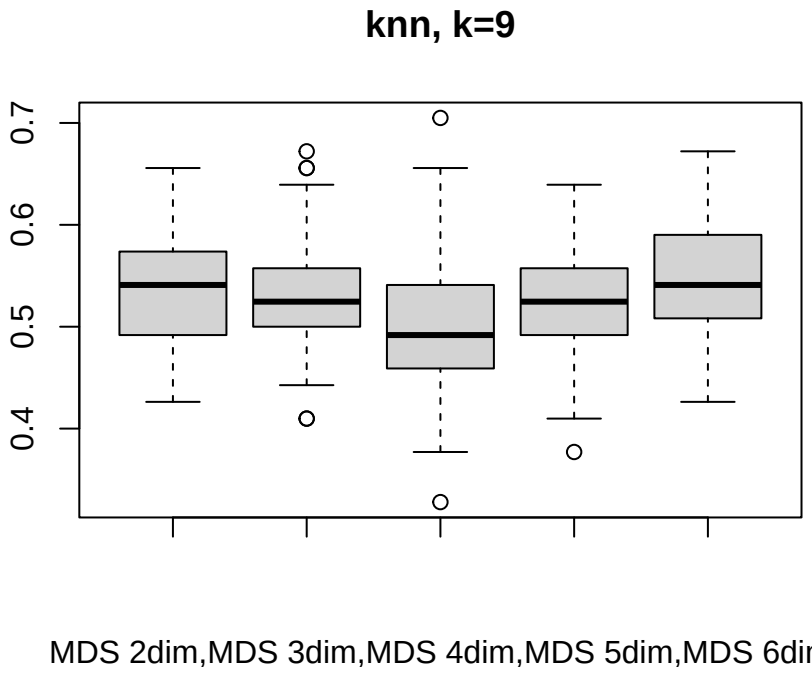
## 5.4 Zastosowanie metod redukcji wymiaru przy klasyfikacji

Metod redukcji wymiaru możemy także użyć w celu poprawy klasyfikacji. W szczególności, bardzo ciekawym zastosowaniem jest użycie metody $MDS$ na danych mieszanego typu, która zwraca nam dane liczbowe po to, żeby następnie użyć otrzymanych wyników do implementacji metod, które na wejściu przyjmują tylko zmienne ilościowe.

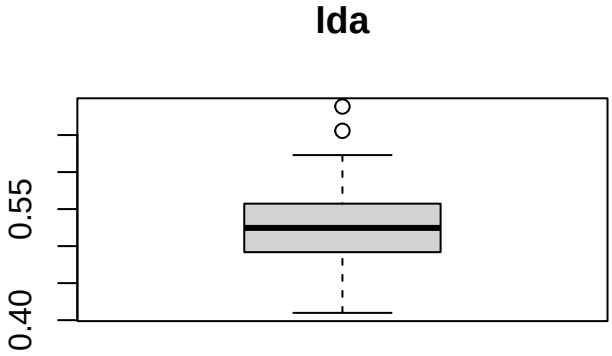
Przeprowadzliśmy symulacje dla metody $k$-najbliższych sąsiadów dla $k = 5$ oraz $k = 9$, dla metody $LDA$, oraz dla metody $KDA$. Użyliśmy skalowania wielowymiarowego do różnych wartości wymiarów, a nastepnie obliczyliśmy dokładność dla klasyfikacji zmiennej `symboling` i wyniki zostały przedstawione za pomocą wykresów pudełkowych na rysunkach 109, 110, 111 oraz 112. UWAGA:redukcję wymiaru stosujemy dla całych danych (zbiór tereningowy + zbiór testowy)!



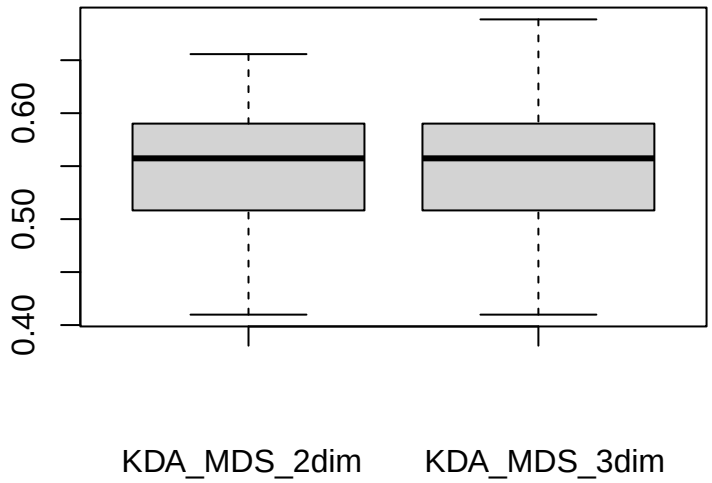

Rysunek 109: Porównanie metody MDS dla różnych wymiarów dla KNN z k=5




Rysunek 110: Porównanie metody MDS dla różnych wymiarów dla KNN z k=9




Rysunek 111: Metoda MDS dla LDA




Rysunek 112: Metoda MDS dla KDA

Niestety, wyniki naszych symmulacji nie wypadły pomyślnie, na wszystkich wykresach powyżej widzimy, że wartości dokładności nie są zbyt dobre, wahają się one między 0.45, a 0.6, co daje gorsze rezultaty niż w przypadku użycia tych samych metod, bez wcześniejszego użycia metody redukcji wymiaru.

# 6 Podsumowanie drugiej części projektu

Na początku drugiej części projektu, w celu klasteryzacji użyliśmy metod grupujących $k - means$ i $PAM$, które zwracały sensowne podziały na partycje, jednakże nie miały one prawie nic wspólnego ze zmienną objaśnianą `symboling`. Algorytm $DBSCAN$ wypadł w naszym porównaniu prawdpodobnie najgorzej, przez niejednorodną gęstość punktów, albo nie umiał wychwycić różnych skupien, albo złączał je w jedną (za) dużą całość. Nie było nawet sensu porównywać ze zmienną `symboling`. Najlepsze rezultaty otrzymaliśmy za pomocą metod hierarchicznych, zwłaszcza tych wprowadzonych na 4-elementowym podzbiorze cech składających się ze zmiennych `normalized.losses`, `fuel.system`, `num.of.doors` i `length`.

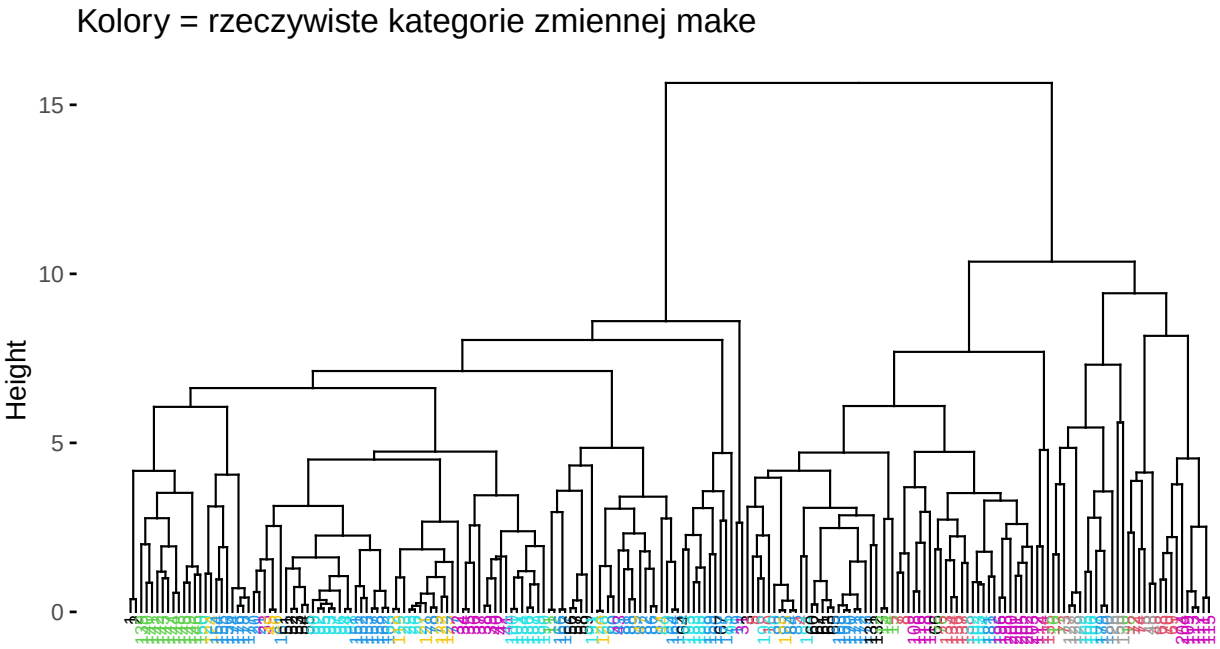
Później zajeliśmy się redukcją wymiaru, która również dała nam kilka ciekawych wyników, przede wszystkim mogliśmy zwizualizować dużo więcej rzeczy przez zwinięcie wielu cech do wymiaru $k = 2$, w szczególności ciekawe rezultaty widzieliśmy porównując $MDS$ dla wymiaru $k = 2$ z etykietkami różnych zmiennych jakościowych, w tym dla nas najważniejszej - `symboling`.

Następnie postaraliśmy się o wprowadzenie metod redukcji przy klasteryzacji oraz klasyfikacji. W przypadku grupowania, powiedzmy, że uzyskaliśmy jakies sensowne wyniki, 2 otrzymane klastry różniły się pod względem charakterystyki, patrząc na różnice w rozkładach różnych zmiennych między partycjami. Natomiast w przypadku klasyfikacji, możemy mówić o rozczarowaniu, ponieważ wprowadzenie metod redukcji wymiaru, tylko pogorszyło dokładność klasyfikacji zmiennej `symboling`.

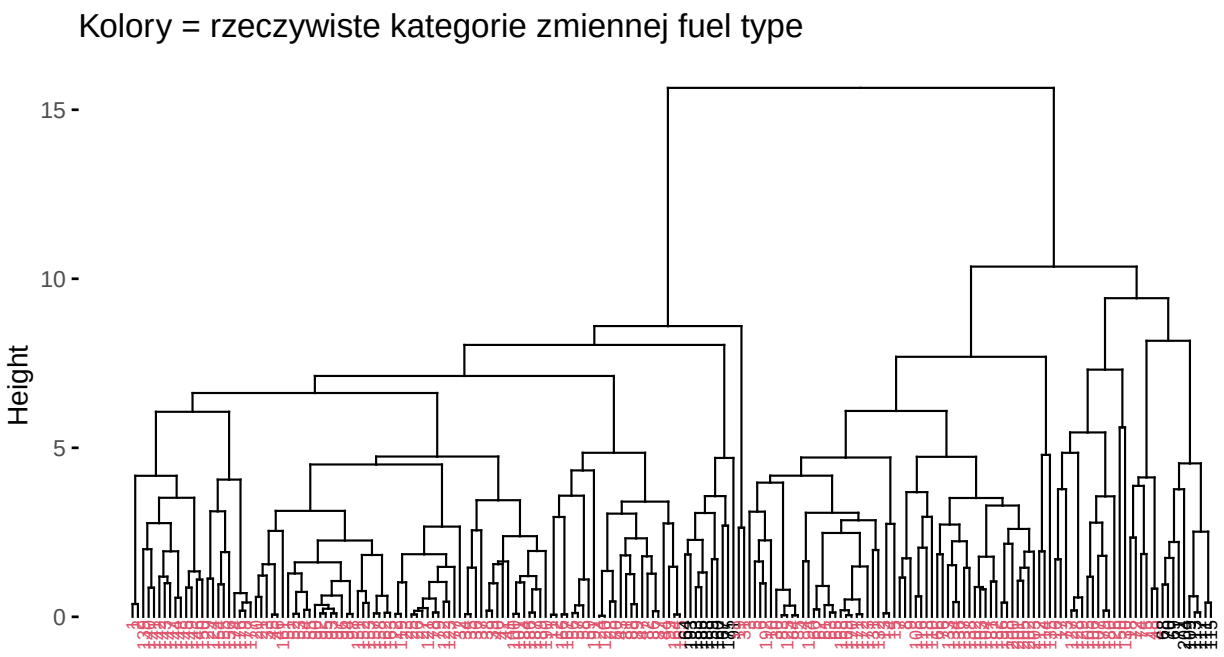
@

# 7 Dodatek

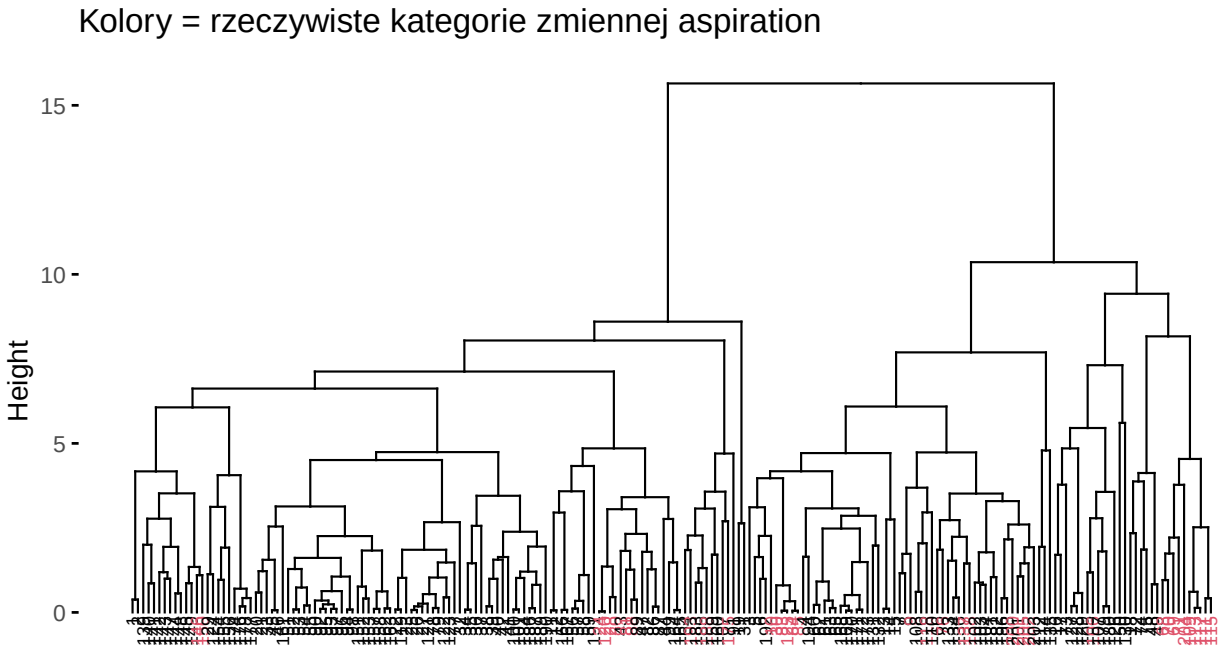
## 7.1 AGNES complete linkage - porównania z pozostałymi zmiennymi jakościowymi



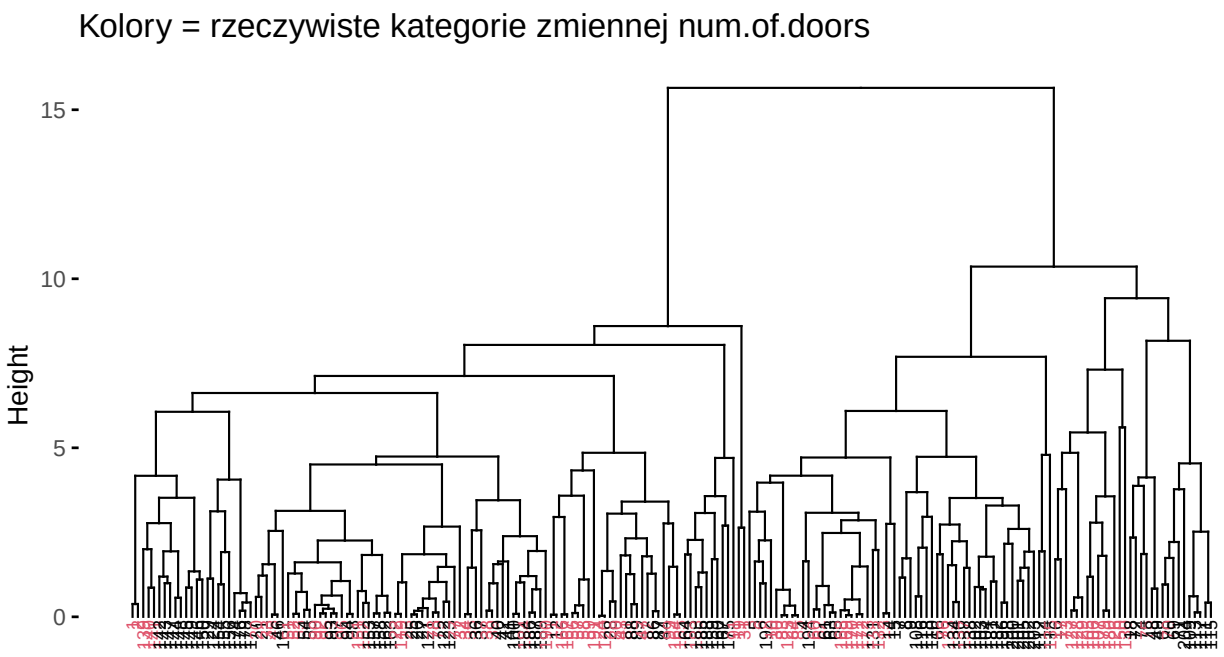

Rysunek 113: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej make



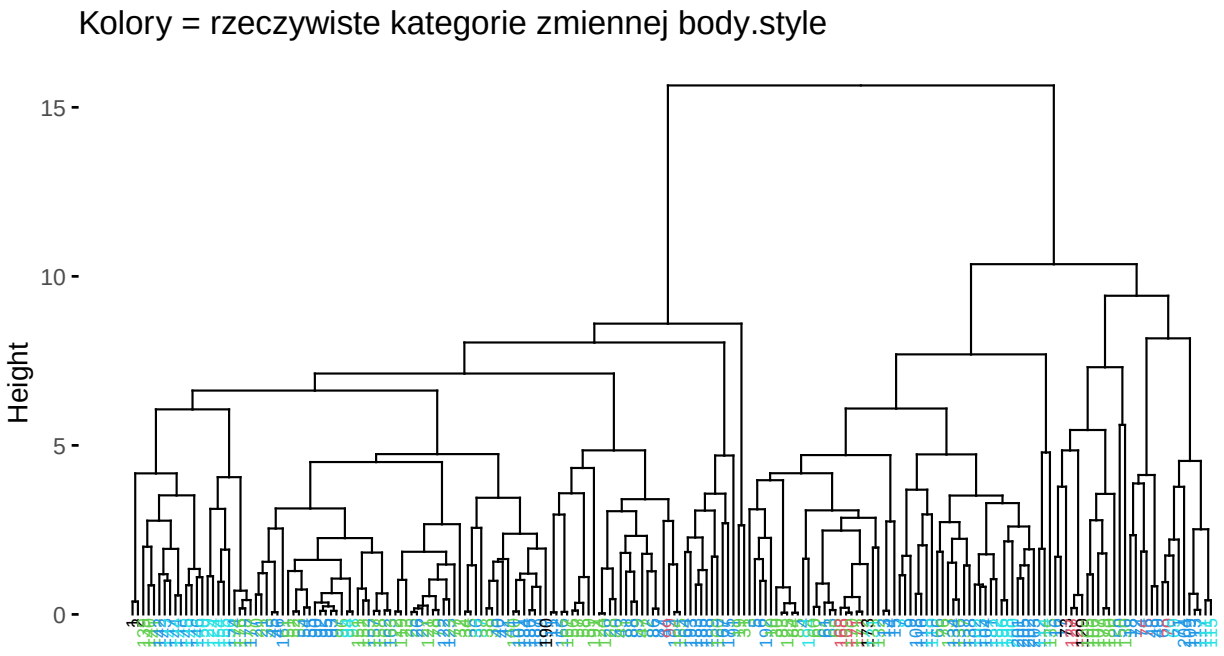

Rysunek 114: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej fuel.type



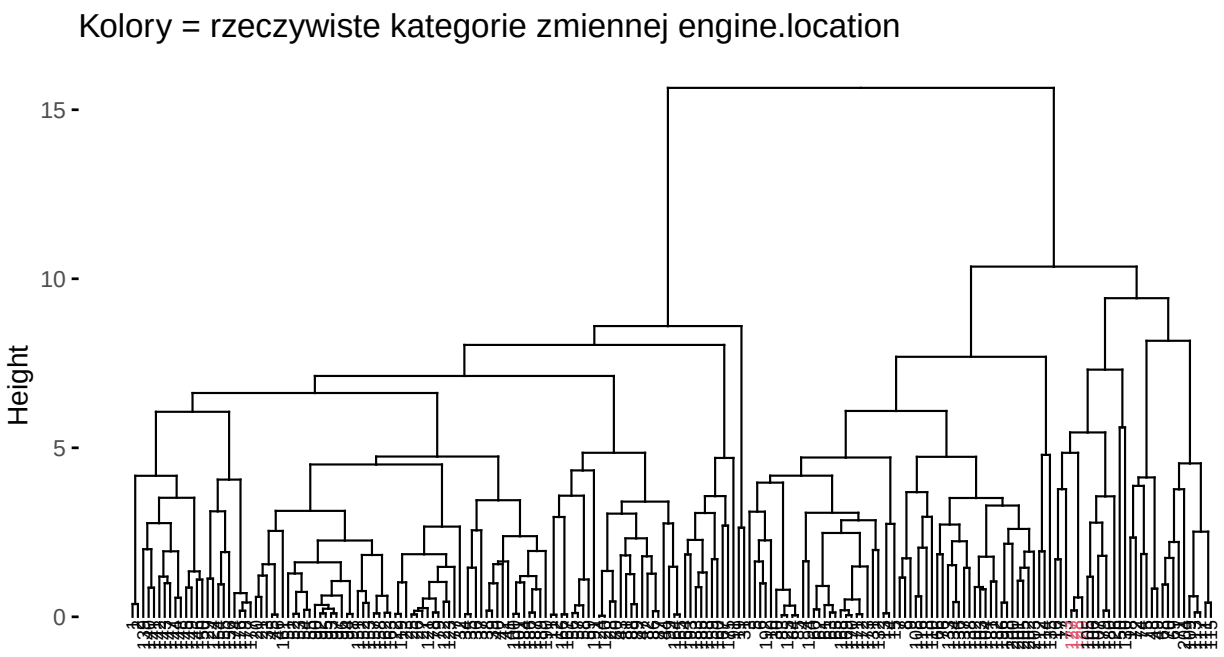

Rysunek 115: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej aspiration



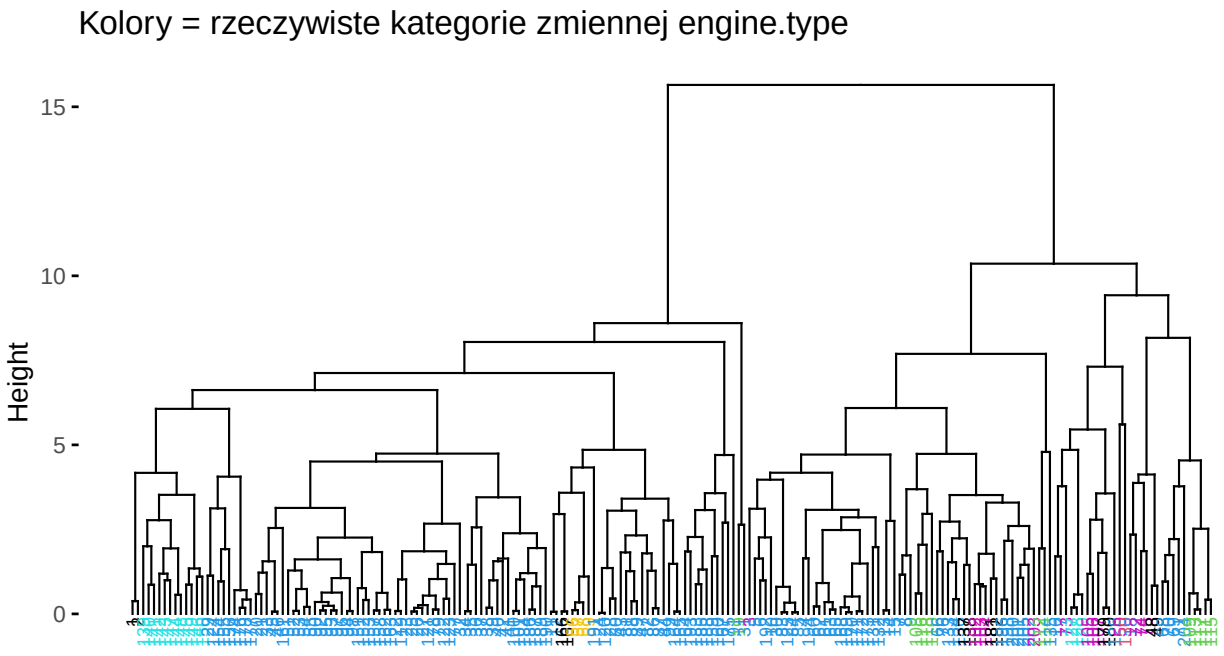

Rysunek 116: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej num.of.doors




Rysunek 117: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej body.style

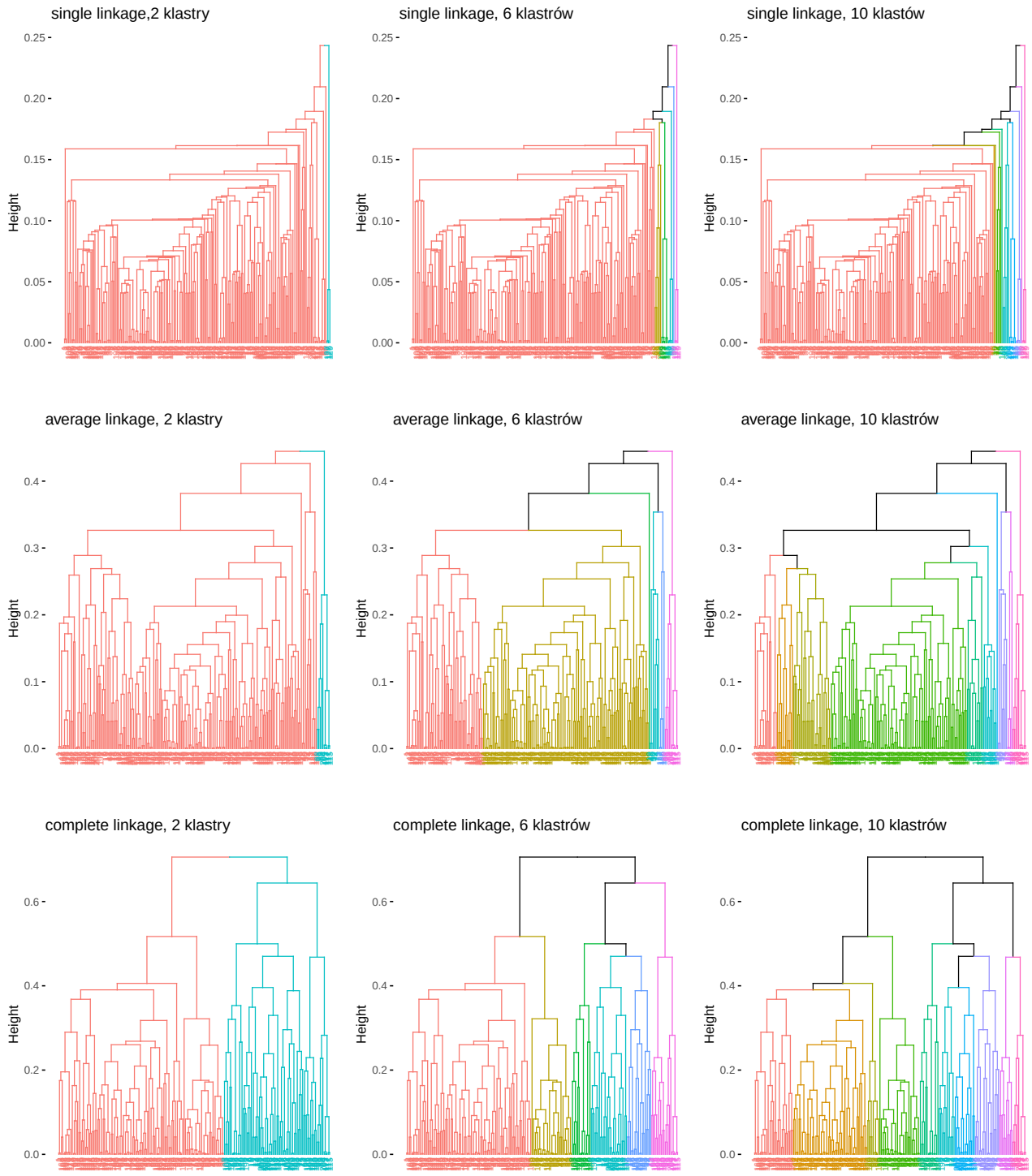



Rysunek 118: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej engine.location




Rysunek 119: dendrogram dla metody AGNES dla zmiennych ilościowych z zaznaczeniem różnych kategorii zmiennej engine.type

## 7.2 dendrogramy AGNES - wszystkie zmienne




Rysunek 120: Porównanie wartości wskaźnika connectivity

# Literatura

[1] https://en.wikipedia.org/wiki/Cram%C3%A9r%27s_V

[2] https://www.rdocumentation.org/packages/rcompanion/versions/2.3.26/topics/cramerV

[3] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/lm

[4] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/anova

[5] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kruskal.test

[6] https://en.wikipedia.org/wiki/Effect_size

[7] https://resources.nu.edu/statsresources/eta

[8] https://cran.r-project.org/web/packages/effectsize/vignettes/interpret.html

[9] https://support.minitab.com/en-us/minitab/21/help-and-how-to/statistics/nonparametrics/how-to/kruskal-wallis-test/interpret-the-results/key-results/

[10] https://stats.stackexchange.com/questions/197827/how-to-interpret-mean-decrease-in-accuracy-and-mean-decrease-gini-in-random-fore

[11] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kmeans

[12] https://en.wikipedia.org/wiki/Silhouette_(clustering)

[13] https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz_nbclus

[14] https://www.rdocumentation.org/packages/cluster/versions/2.1.4/topics/daisy

[15] https://medium.com/analytics-vidhya/gowers-distance-899f9c4bd553

[16] https://cran.r-project.org/web/packages/factoextra/factoextra.pdf

[17] https://en.wikipedia.org/wiki/Dunn_index

[18] https://en.wikipedia.org/wiki/DBSCAN

[19] https://www.rdocumentation.org/packages/gplots/versions/3.1.3/topics/heatmap.2

[20] https://en.wikipedia.org/wiki/Hierarchical_clustering

[21] https://www.rdocumentation.org/packages/cluster/versions/2.1.4/topics/agnes

[22] https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz_dend

[23] https://pl.wikipedia.org/wiki/Analiza_g%C5%82%C3%B3wnych_sk%C5%82adowych

[24] https://www.rdocumentation.org/packages/FactoMineR/versions/2.8/topics/PCA

[25] https://en.wikipedia.org/wiki/Multidimensional_scaling

[26] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/cmdscale