

Błażej Domagała - WED

Lista 1

Zadanie 1

a) Nazwa zbioru: Wine

```
> library(datasets)
>
> path = "C:\\users\\petioff\\Desktop\\repos\\uo\\rok 3\\wprowadzenie do eksploracji danych\\lista1\\zadanie2" # używając podwójnych ukośników
> setwd(path) ## ustawienie ścieżki
>
> ## zmiana nazwy kolumn:
>
> # Załadowanie danych
> # Spróbuj wczytać dane z innym separatorem
> wine <- read.csv('wine\\wine.data', header=FALSE)
```

b) Krótki tekstowy opis zbioru

Zbiór danych "Wine" zawiera wyniki analizy chemicznej win wyprodukowanych w określonym regionie we Włoszech przez trzech różnych producentów. Analiza chemiczna dotyczy 13 różnych składników zawartych w winach.

```
> ## zmiana nazwy kolumn:
>
> # Załadowanie danych
> # Spróbuj wczytać dane z innym separatorem
> wine <- read.csv('wine\\wine.data', header=FALSE)
>
> # zmień nazwy kolumn
> names(wine) <- c('class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline')
>
> ##
```

c) Liczba obserwacji w zbiorze: 178

```
> view(wine)
>
> ## Liczba obserwacji w zbiorze
> nrow(wine)
[1] 178
>
```

d) Liczba kolumn: 14 (13 atrybutów + 1 kolumna identyfikująca klasę)

```
> ## Liczba kolumn
> print(names(wine))
[1] "class"                "Alcohol"              "Malic acid"           "Ash"
[5] "Alcalinity of ash"    "Magnesium"            "Total phenols"        "Flavanoids"
[9] "Nonflavanoid phenols" "Proanthocyanins"      "Color intensity"      "Hue"
[13] "OD280/OD315 of diluted wines" "Proline"
```

e) Zmienna celu:

- Nazwa kolumny z klasą: Class
- Liczba klas: 3 (Klasy 1, 2 i 3)

```
> ## Zmienna celu
> unique(wine$class)
[1] 1 2 3
>
```

f) Wykaz i opis cech:

1. **Class:** Zmienna katégoryczna. Klasa wina.
2. **Alcohol:** Zmienna ilościowa. Zawartość alkoholu.
3. **Malic acid:** Zmienna ilościowa. Zawartość kwasu jabłkowego.
4. **Ash:** Zmienna ilościowa. Zawartość popiołu.
5. **Alcalinity of ash:** Zmienna ilościowa. Zasadowość popiołu.
6. **Magnesium:** Zmienna ilościowa. Zawartość magnezu.
7. **Total phenols:** Zmienna ilościowa. Całkowita zawartość fenoli.
8. **Flavanoids:** Zmienna ilościowa. Zawartość flawonoidów.
9. **Nonflavanoid phenols:** Zmienna ilościowa. Zawartość fenoli nieflawonoidowych.
10. **Proanthocyanins:** Zmienna ilościowa. Zawartość proantocyjanidyn.
11. **Color intensity:** Zmienna ilościowa. Intensywność koloru.
12. **Hue:** Zmienna ilościowa. Odcień.
13. **OD280/OD315 of diluted wines:** Zmienna ilościowa. Stosunek absorbancji przy 280 nm do 315 nm w rozcieńczonych winach.
14. **Proline:** Zmienna ilościowa. Zawartość prolina.

```
> summary(wine)
  class      Alcohol      Malic acid      Ash      Alcalinity of ash      Magnesium      Total phenols      Flavanoids      Nonflavanoid phenols
Min.   :1.000   Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60   Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300
1st Qu.:1.000   1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20   1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700
Median :2.000   Median :13.05   Median :1.865   Median :2.360   Median :19.50   Median : 98.00   Median :2.355   Median :2.135   Median :0.3400
Mean   :1.938   Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49   Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619
3rd Qu.:3.000   3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50   3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375
Max.   :3.000   Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00   Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600
Proanthocyanins color intensity      Hue      OD280/OD315 of diluted wines      Proline
Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270   Min.   : 278.0
1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938   1st Qu.: 500.5
Median :1.555   Median : 4.690   Median :0.9650   Median :2.780   Median : 673.5
Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612   Mean   : 746.9
3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0
Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000   Max.   :1680.0
```

Zadanie 2

- a) Zmiana nazw kolumn (pierwszą kolumnę – zmienną celu – proszę nazwać „Class”; nazwy pozostałych kolumn – atrybutów – są podane w pliku „wine.names”)

```
R 4.3.1 C:/Users/petitoff/Desktop/repos/UO/rok 3/Wprowadzenie do eksploracji danych/lista1/zadanie2
> library(datasets)
>
> path = "C:\\Users\\petitoff\\Desktop\\repos\\UO\\rok 3\\Wprowadzenie do eksploracji danych\\lista1\\zadanie2" # używając podwójnych ukośników
> setwd(path) ## ustawienie ścieżki
>
> ## zmiana nazwy kolumn:
>
> # Załadowanie danych
> wine <- read.csv("wine\\wine.data", header=FALSE)
>
> # Zmień nazwy kolumn
> names(wine) <- c('Class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline')
>
```

- b) Polecenie View (fragment print-screena z tabelką)

```
> ## =====
> view(wine)
>
```

RStudio interface showing a dataset named 'wine' with 178 entries and 14 columns. The columns are: Class, Alcohol, Malic acid, Ash, Alkalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline. The data is displayed in a table format.

Class	Alcohol	Malic acid	Ash	Alkalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.640000	1.040	3.92	
2	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.380000	1.050	3.40	
3	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.680000	1.030	3.17	
4	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.800000	0.860	3.45	
5	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.320000	1.040	2.93	
6	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.750000	1.050	2.85	
7	14.39	1.87	2.45	14.6	96	2.50	2.52	0.30	1.98	5.250000	1.020	3.58	
8	14.06	2.15	2.61	17.6	121	2.60	2.51	0.31	1.25	5.050000	1.060	3.58	
9	14.83	1.64	2.17	14.0	97	2.80	2.98	0.29	1.98	5.200000	1.080	2.85	
10	13.86	1.35	2.27	16.0	98	2.98	3.15	0.22	1.85	7.220000	1.010	3.55	
11	14.10	2.16	2.30	18.0	105	2.95	3.32	0.22	2.38	5.750000	1.250	3.17	
12	14.12	1.48	2.32	16.8	95	2.20	2.43	0.26	1.57	5.000000	1.170	2.82	
13	13.75	1.73	2.41	16.0	89	2.60	2.76	0.29	1.81	5.600000	1.150	2.90	
14	14.75	1.73	2.39	11.4	91	3.10	3.69	0.43	2.81	5.400000	1.250	2.73	
15	14.38	1.87	2.38	12.0	102	3.30	3.64	0.29	2.96	7.500000	1.200	3.00	
16	13.63	1.81	2.70	17.2	112	2.85	2.91	0.30	1.46	7.300000	1.280	2.88	
17	14.30	1.92	2.72	20.0	120	2.80	3.14	0.33	1.97	6.200000	1.070	2.65	
18	13.83	1.57	2.62	20.0	115	2.95	3.40	0.40	1.72	6.600000	1.130	2.57	
19	14.19	1.59	2.48	16.5	108	3.30	3.93	0.32	1.86	8.700000	1.230	2.82	
20	13.64	3.10	2.56	15.2	116	2.70	3.03	0.17	1.66	5.100000	0.960	3.36	
21	14.06	1.63	2.28	16.0	126	3.00	3.17	0.24	2.10	5.650000	1.090	3.71	
22	12.93	3.80	2.65	18.6	102	2.41	2.41	0.25	1.98	4.500000	1.030	3.52	
23	13.71	1.86	2.36	16.6	101	2.61	2.88	0.27	1.69	3.800000	1.110	4.00	
24	12.85	1.60	2.52	17.8	95	2.48	2.37	0.26	1.46	3.930000	1.090	3.63	
25	13.50	1.81	2.61	20.0	96	2.53	2.61	0.28	1.66	3.520000	1.120	3.82	
26	13.05	2.05	3.22	25.0	124	2.63	2.68	0.47	1.92	3.580000	1.130	3.20	
27	13.30	1.77	2.62	16.1	93	2.85	2.94	0.34	1.45	4.800000	0.920	3.72	

Showing 1 to 27 of 178 entries, 14 total columns

Console Terminal Background Jobs

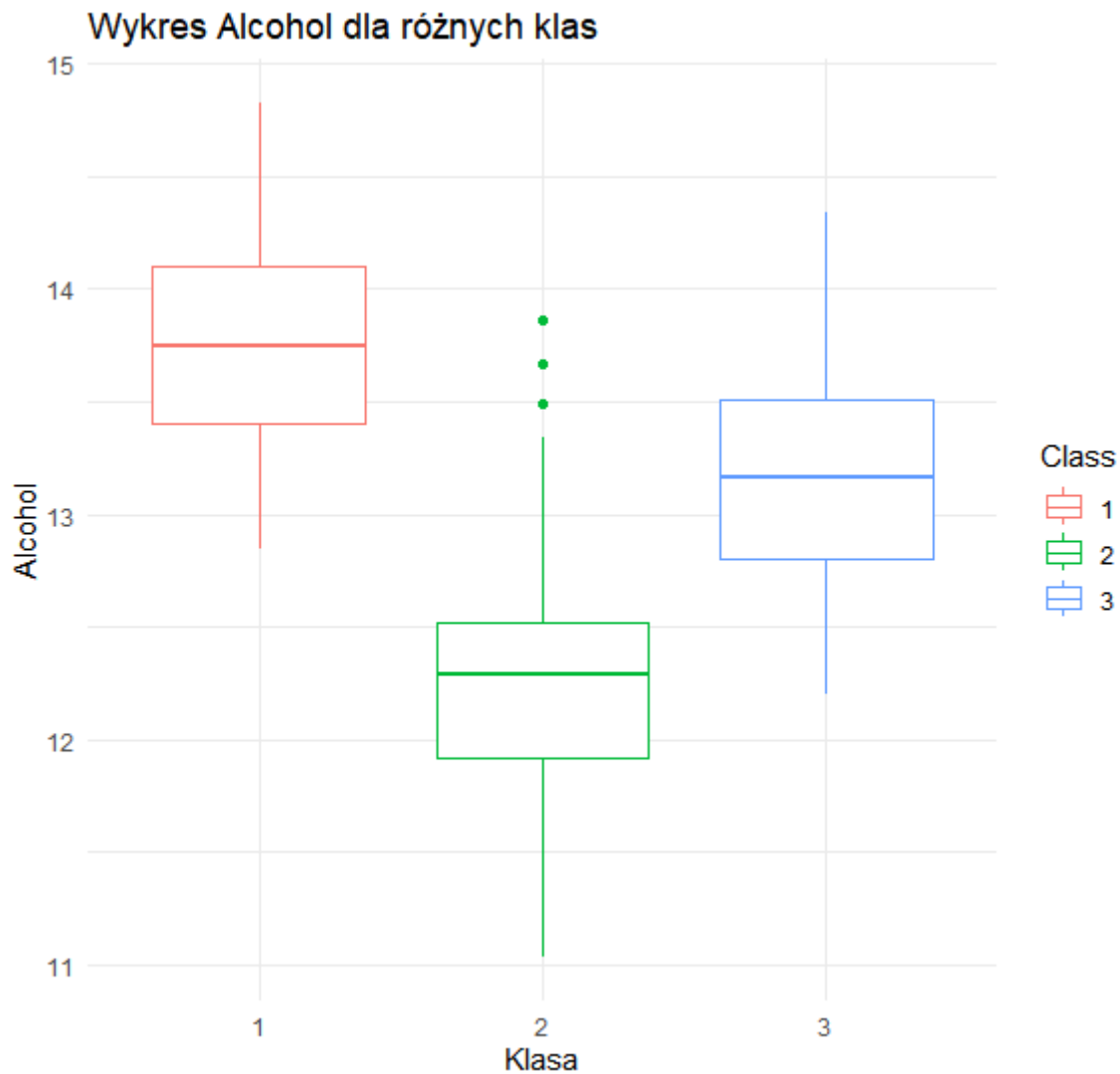
```
R 4.3.1 C:\Users\petitoff\Desktop\repos\UO\rok 3\Wprowadzenie do eksploracji danych\lista1\zadanie2/
> library(datasets)
>
> path = "C:\\Users\\petitoff\\Desktop\\repos\\UO\\rok 3\\Wprowadzenie do eksploracji danych\\lista1\\zadanie2" # używając podwójnych ukośników
> setwd(path) ## ustawienie ścieżki
>
> ## zmiana nazwy kolumn:
```

c) Podsumowanie cech (summary)

```
> ## Podsumowanie cech
> summary(wine)
      Class      Alcohol      Malic acid      Ash      Alkalinity of ash      Magnesium      Total phenols      Flavanoids      Nonflavanoid phenols
Min.   :1.000   Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60   Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300
1st Qu.:1.000   1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20   1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700
Median :2.000   Median :13.05   Median :1.865   Median :2.360   Median :19.50   Median : 98.00   Median :2.355   Median :2.135   Median :0.3400
Mean   :1.938   Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49   Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619
3rd Qu.:3.000   3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50   3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375
Max.   :3.000   Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00   Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600

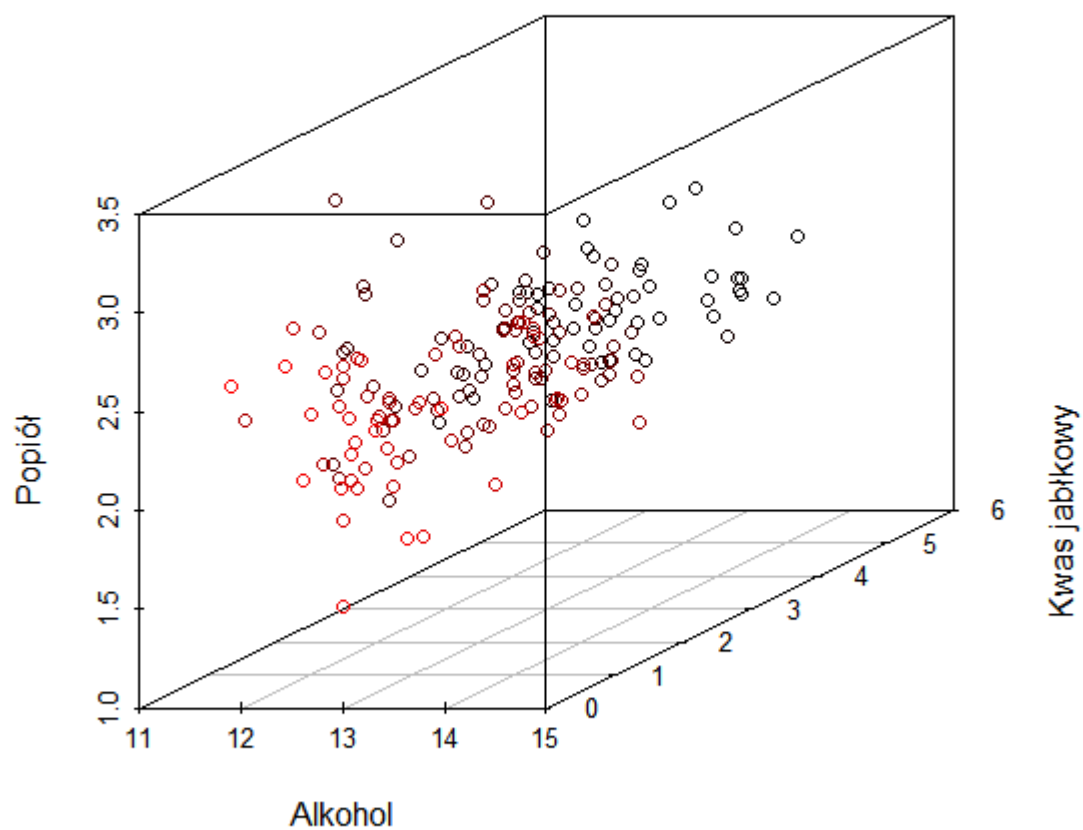
      Proanthocyanins      Color intensity      Hue      OD280/OD315 of diluted wines      Proline
Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270   Min.   : 278.0
1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938   1st Qu.: 500.5
Median :1.555   Median : 4.690   Median :0.9650   Median :2.780   Median : 673.5
Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612   Mean   : 746.9
3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0
Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000   Max.   :1680.0
```

d) Wykres 2D ilustrujący wybraną cechę dla różnych klas



```
> ## wykres 2D:
> data(wine, package = "datasets")
Komunikat ostrzegawczy:
W poleceniu 'data(wine, package = "datasets")':
zbiór danych 'wine' nie został znaleziony
> # Załadowanie biblioteki do tworzenia wykresów
> library(ggplot2)
> # wybór cechy do przedstawienia na wykresie - np. "Alcohol"
> feature <- "Alcohol"
> # konwersja zmiennej 'class' na faktor
> wine$class <- as.factor(wine$class)
> # Stworzenie wykresu 2D z użyciem ggplot2
> p <- ggplot(wine, aes(x = class, y = !!sym(feature), color = class)) +
+   geom_boxplot() +
+   labs(title = paste("wykres", feature, "dla różnych klas"),
+        x = "Klasa",
+        y = feature) +
+   theme_minimal()
> # Wyświetlenie wykresu
> print(p)
> ## wykres 3D:
> library(scatterplot3d)
> # Stwórz wykres 3D dla wybranych cech
> scatterplot3d(wine[,c("Alcohol", "Malic acid", "Ash",
+   xlab="Alcohol", ylab="Kwas jabłkowy", zlab="Popiół",
+   highlight.3d=TRUE, angle=30)
> # Zapis do pliku
> ggsave("C:\\Users\\petitoff\\desktop\\repos\\00\\rok 3\\wprowadzenie do eksploracji danych\\lista1\\zadanie2\\wykres.png", plot = p, width = 10, height = 6, dpi = 300)
>
```

e) Wykres 3D dla trzech wybranych cech (bez klasy)



```
> # Stwórz wykres 3D dla wybranych cech
> scatterplot3d(wine$Alcohol, wine$`Malic acid`, wine$Ash,
+               xlab="Alkohol", ylab="Kwas jabłkowy", zlab="Popiół",
+               highlight.3d=TRUE, angle=30)
> |
```

Lista 2

- a) Dla wybranej cechy wyświetlić wybrane wartości stosując: zakres wartości, sekwencję indeksów (np. co dziesiąty indeks), indeksy ujemne, warunki logiczne.

Kod:

```
path = "C:\\Users\\petitoff\\Desktop\\repos\\UO\\rok 3\\Wprowadzenie do eksploracji  
danych\\lista2"
```

```
setwd(path) ## ustawienie ścieżki
```

```
# Załadowanie danych
```

```
wine <- read.csv('wine\\wine.data', header = FALSE)
```

```
# Zmień nazwy kolumn
```

```
names(wine) <-
```

```
c(
```

```
  'Class',
```

```
  'Alcohol',
```

```
  'Malic acid',
```

```
  'Ash',
```

```
  'Alcalinity of ash',
```

```
  'Magnesium',
```

```
  'Total phenols',
```

```
  'Flavanoids',
```

```
  'Nonflavanoid phenols',
```

```
  'Proanthocyanins',
```

```
  'Color intensity',
```

```
  'Hue',
```

```
  'OD280/OD315 of diluted wines',
```

```
  'Proline'
```

```
)
```

a) Dla wybranej cechy wyświetlić wybrane wartości stosując: zakres wartości, sekwencję indeksów (np. co dziesiąty indeks), indeksy ujemne, warunki logiczne.

```
# Zakres wartości
```

```
wine$Alcohol[5:15]
```

```
# Sekwencja indeksów
```

```
wine$Alcohol[seq(1, nrow(wine), 10)]
```

```
# Indeksy ujemne
```

```
wine$Alcohol[-(1:5)]
```

```
# Warunki logiczne
```

```
wine$Alcohol[wine$Alcohol > 14]
```

```
> path = "C:\\Users\\petitoff\\Desktop\\repos\\U0\\rok 3\\wprowadzenie do eksploracji danych\\lista2"
> setwd(path) ## ustawienie ścieżki
>
> # Załadowanie danych
> wine <- read.csv('wine\\wine.data', header=FALSE)
>
> # Zmień nazwy kolumn
> names(wine) <- c('Class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoi
ds', 'Nonflavanoid phenols', 'Proanthocyanins', 'color intensity', 'Hue', 'od280/od315 of diluted wines', 'Proline')
>
> # head(wine)
>
> # a) Dla wybranej cechy wyświetlić wybrane wartości stosując: zakres wartości, sekwencję indeksów (np. co dziesiąty i
ndeks), indeksy ujemne, warunki logiczne.
> # Zakres wartości
> wine$Alcohol[5:15]
[1] 13.24 14.20 14.39 14.06 14.83 13.86 14.10 14.12 13.75 14.75 14.38
>
> # Sekwencja indeksów
> wine$Alcohol[seq(1, nrow(wine), 10)]
[1] 14.23 14.10 14.06 13.73 13.56 13.05 12.33 12.29 12.00 12.08 12.08 11.46 11.45 12.86 12.93 13.50 12.36 12.20
>
> # Indeksy ujemne
> wine$Alcohol[-(1:5)]
[1] 14.20 14.39 14.06 14.83 13.86 14.10 14.12 13.75 14.75 14.38 13.63 14.30 13.83 14.19 13.64 14.06 12.93 13.71
[19] 12.85 13.50 13.05 13.39 13.30 13.87 14.02 13.73 13.58 13.68 13.76 13.51 13.48 13.28 13.05 13.07 14.22 13.56
[37] 13.41 13.88 13.24 13.05 14.21 14.38 13.90 14.10 13.94 13.05 13.83 13.82 13.77 13.74 13.56 14.22 13.29 13.72
[55] 12.37 12.33 12.64 13.67 12.37 12.17 12.37 13.11 12.37 13.34 12.21 12.29 13.86 13.49 12.99 11.96 11.66 13.03
[73] 11.84 12.33 12.70 12.00 12.72 12.08 13.05 11.84 12.67 12.16 11.65 11.64 12.08 12.08 12.00 12.69 12.29 11.62
[91] 12.47 11.81 12.29 12.37 12.29 12.08 12.60 12.34 11.82 12.51 12.42 12.25 12.72 12.22 11.61 11.46 12.52 11.76
[109] 11.41 12.08 11.03 11.82 12.42 12.77 12.00 11.45 11.56 12.42 13.05 11.87 12.07 12.43 11.79 12.37 12.04 12.86
[127] 12.88 12.81 12.70 12.51 12.60 12.25 12.53 13.49 12.84 12.93 13.36 13.52 13.62 12.25 13.16 13.88 12.87 13.32
[145] 13.08 13.50 12.79 13.11 13.23 12.58 13.17 13.84 12.45 14.34 13.48 12.36 13.69 12.85 12.96 13.78 13.73 13.45
[163] 12.82 13.58 13.40 12.20 12.77 14.16 13.71 13.40 13.27 13.17 14.13
>
> # Warunki logiczne
> wine$Alcohol[wine$Alcohol > 14]
[1] 14.23 14.37 14.20 14.39 14.06 14.83 14.10 14.12 14.75 14.38 14.30 14.19 14.06 14.02 14.22 14.21 14.38 14.10
[19] 14.22 14.34 14.16 14.13
> |
```

b) Wyświetlić wybrane wiersze i kolumny z tabeli.

```
# b) Wybranie wierszy i kolumn:
```

```
selected_rows_columns <- wine[1:10, c("Alcohol", "Malic acid")]
```

```
print(selected_rows_columns)
```

```
# indeksy
```

wine[5:15,]

```
> # b) wybranie wierszy i kolumn:
> selected_rows_columns <- wine[1:10, c("Alcohol", "Malic acid")]
> print(selected_rows_columns)
  Alcohol Malic acid
1    14.23      1.71
2    13.20      1.78
3    13.16      2.36
4    14.37      1.95
5    13.24      2.59
6    14.20      1.76
7    14.39      1.87
8    14.06      2.15
9    14.83      1.64
10   13.86      1.35
>
> # indeksy
> wine[5:15, ]
  Class Alcohol Malic acid Ash Alkalinity of ash Magnesium Total phenols Flavonoids Nonflavanoid phenols
5      1   13.24      2.59 2.87          21.0      118          2.80      2.69          0.39
6      1   14.20      1.76 2.45          15.2      112          3.27      3.39          0.34
7      1   14.39      1.87 2.45          14.6       96          2.50      2.52          0.30
8      1   14.06      2.15 2.61          17.6      121          2.60      2.51          0.31
9      1   14.83      1.64 2.17          14.0       97          2.80      2.98          0.29
10     1   13.86      1.35 2.27          16.0       98          2.98      3.15          0.22
11     1   14.10      2.16 2.30          18.0      105          2.95      3.32          0.22
12     1   14.12      1.48 2.32          16.8       95          2.20      2.43          0.26
13     1   13.75      1.73 2.41          16.0       89          2.60      2.76          0.29
14     1   14.75      1.73 2.39          11.4       91          3.10      3.69          0.43
15     1   14.38      1.87 2.38          12.0      102          3.30      3.64          0.29
  Proanthocyanins Color intensity Hue OD280/OD315 of diluted wines Proline
5          1.82          4.32 1.04          2.93      735
6          1.97          6.75 1.05          2.85     1450
7          1.98          5.25 1.02          3.58     1290
8          1.25          5.05 1.06          3.58     1295
9          1.98          5.20 1.08          2.85     1045
10         1.85          7.22 1.01          3.55     1045
11         2.38          5.75 1.25          3.17     1510
12         1.57          5.00 1.17          2.82     1280
13         1.81          5.60 1.15          2.90     1320
14         2.81          5.40 1.25          2.73     1150
15         2.96          7.50 1.20          3.00     1547
> |
```

c) Dodać do tabeli nową kolumnę z wartościami obliczonymi na podstawie innych wybranych kolumn.

c) Dodanie nowej kolumny:

Sprawdzanie, czy kolumna istnieje i zawiera dane

```
if ("Total phenols" %in% names(wine) &&
```

```
    !all(is.na(wine$`Total phenols`))) {
```

```
  # Dodanie nowej kolumny
```

```
  wine$Total.phenols.squared <- wine$`Total phenols` ^ 2
```

```
  head(wine)
```

```
} else {
```

```
  cat("Column 'Total phenols' does not exist or is empty.")
```

```
}
```



```

> # c) Dodanie nowej kolumny:
> # Sprawdzanie, czy kolumna istnieje i zawiera dane
> if ("Total phenols" %in% names(wine) && !all(is.na(wine$`Total phenols`))) {
+ # Dodanie nowej kolumny
+ wine$Total.phenols.squared <- wine$`Total phenols`^2
+ head(wine)
+ } else {
+ cat("Column 'Total phenols' does not exist or is empty.")
+ }
  Class Alcohol Malic acid Ash Alkalinity of ash Magnesium Total phenols Flavanoids Nonflavanoid phenols
1      1    14.23      1.71 2.43          15.6      127      2.80      3.06              0.28
2      1    13.20      1.78 2.14          11.2     100      2.65      2.76              0.26
3      1    13.16      2.36 2.67          18.6     101      2.80      3.24              0.30
4      1    14.37      1.95 2.50          16.8     113      3.85      3.49              0.24
5      1    13.24      2.59 2.87          21.0     118      2.80      2.69              0.39
6      1    14.20      1.76 2.45          15.2     112      3.27      3.39              0.34
  Proanthocyanins Color intensity Hue OD280/OD315 of diluted wines Proline Total.phenols.squared
1          2.29          5.64 1.04          3.92     1065          7.8400
2          1.28          4.38 1.05          3.40     1050          7.0225
3          2.81          5.68 1.03          3.17     1185          7.8400
4          2.18          7.80 0.86          3.45     1480          14.8225
5          1.82          4.32 1.04          2.93      735          7.8400
6          1.97          6.75 1.05          2.85     1450          10.6929
>

```

d) Podać wartości podstawowych statystyk dla wybranej kolumny: zakres, średnia, mediana,

d) Statystyki podstawowe dla wybranej kolumny, np. "Alcohol"

```
cat("Statystyki dla kolumny 'Alcohol':", "\n")
```

```
cat("Zakres: ", min(wine$Alcohol), " - ", max(wine$Alcohol), "\n")
```

```
cat("Średnia: ", mean(wine$Alcohol), "\n")
```

```
cat("Mediana: ", median(wine$Alcohol), "\n")
```

```
cat("Odchylenie standardowe: ", sd(wine$Alcohol), "\n")
```

```
cat("Kurtoza: ", moments::kurtosis(wine$Alcohol), "\n")
```

```
cat("Skośność: ", moments::skewness(wine$Alcohol), "\n")
```

```
cat("Kwantyle: ", quantile(wine$Alcohol, probs = c(0.25, 0.5, 0.75)), "\n")
```

```

> # d) Statystyki podstawowe dla wybranej kolumny, np. "Alcohol"
> cat("Statystyki dla kolumny 'Alcohol':", "\n")
Statystyki dla kolumny 'Alcohol':
> cat("Zakres: ", min(wine$Alcohol), " - ", max(wine$Alcohol), "\n")
Zakres: 11.03 - 14.83
> cat("Średnia: ", mean(wine$Alcohol), "\n")
Średnia: 13.00062
> cat("Mediana: ", median(wine$Alcohol), "\n")
Mediana: 13.05
> cat("odchylenie standardowe: ", sd(wine$Alcohol), "\n")
odchylenie standardowe: 0.8118265
> cat("Kurtoza: ", moments::kurtosis(wine$Alcohol), "\n")
Kurtoza: 2.13774
> cat("skośność: ", moments::skewness(wine$Alcohol), "\n")
skośność: -0.05104747
> cat("Kwantyle: ", quantile(wine$Alcohol, probs = c(0.25, 0.5, 0.75)), "\n")
Kwantyle: 12.3625 13.05 13.6775
>

```

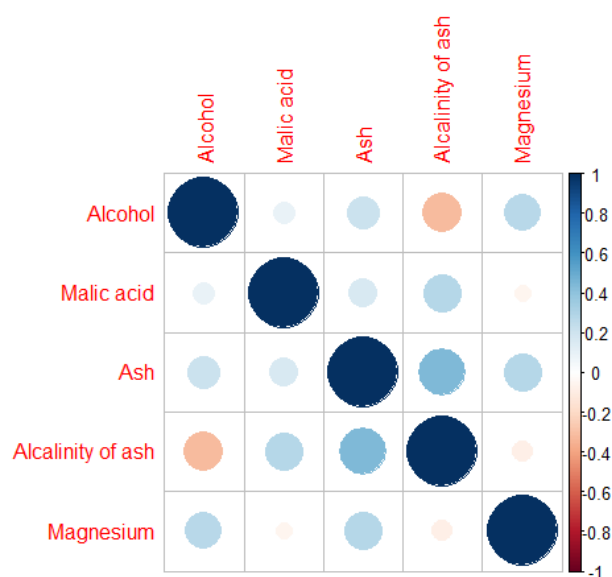
e) Wyznaczyć i zilustrować na wykresie macierz korelacji dla wybranych pięciu zmiennych.

e) Macierz korelacji dla wybranych pięciu zmiennych i jej wizualizacja

```
selected_vars <-
  wine[, c('Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium')]
cor_matrix <- cor(selected_vars)
```

```
library(corrplot)
corrplot(cor_matrix, method = "circle")
```

```
# e) Macierz korelacji dla wybranych pięciu zmiennych i jej wizualizacja
selected_vars <- wine[, c('Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium')]
cor_matrix <- cor(selected_vars)
```



f) Wydrukować histogramy dla trzech różnych zmiennych, przedyskutować wyniki.

f) Histogramy dla trzech różnych zmiennych

```
par(mfrow = c(1, 3)) # ustawienie layoutu na 1 wiersz i 3 kolumny
```

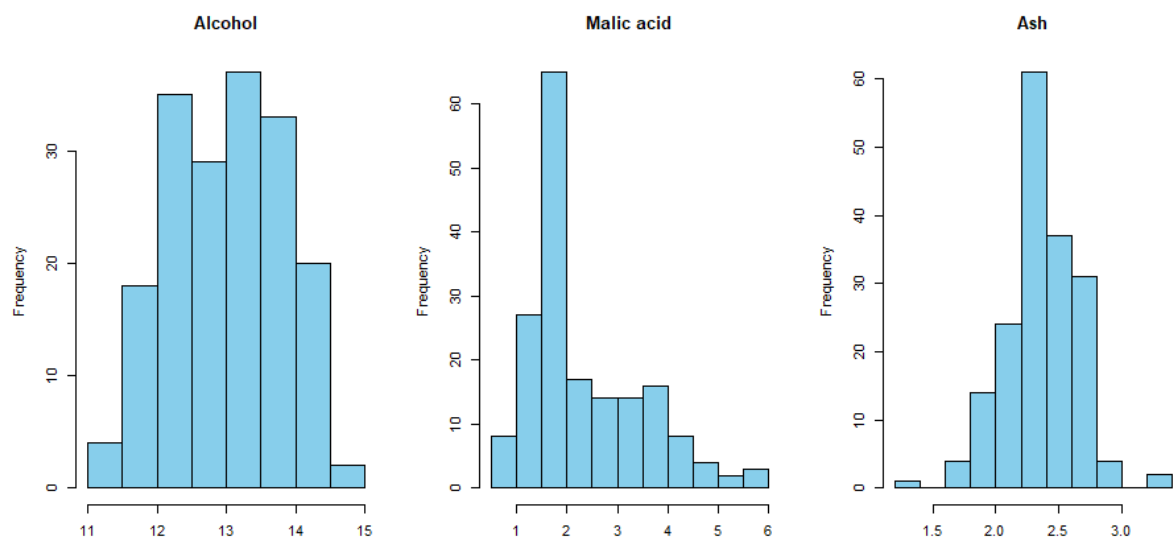
```
hist(wine$Alcohol,
     main = 'Alcohol',
     xlab = "",
     col = 'skyblue')
```

```
hist(
  wine$`Malic acid`,
  main = 'Malic acid',
  xlab = "",
  col = 'skyblue')
```

)

```
hist(wine$Ash,  
      main = 'Ash',  
      xlab = '',  
      col = 'skyblue')
```

```
60  
61 # f) Histogramy dla trzech różnych zmiennych  
62 par(mfrow=c(1,3)) # ustawienie layoutu na 1 wiersz i 3 kolumny  
63 hist(wine$Alcohol, main='Alcohol', xlab='', col='skyblue')  
64 hist(wine$Malic acid, main='Malic acid', xlab='', col='skyblue')  
65 hist(wine$Ash, main='Ash', xlab='', col='skyblue')
```



Lista 3

a) Usuń kolumny z wartościami nominalnymi (identyfikatory, itp.) – jeżeli są, inne niż zmienna celu.

Zakładając, że zmienna celu to V1, a pozostałe kolumny to wartości liczbowe nie ma kolumn z wartościami nominalnymi do usunięcia.

Jeśli jednak byłyby takie kolumny, można by je usunąć za pomocą polecenia subset.

Przykład: Jeśli kolumna V2 była by zmienną nominalną, można by ją usunąć następująco:

```
wine <- subset(wine, select = -V2) # Usuń kolumnę V2 z ramki danych wine
```

```
View(wine) # Wyświetl dane po usunięciu kolumny V2
```

b) Zmień nazwy kolumn na nazwy w języku polskim. Nowe nazwy powinny być: krótkie, znaczące, bez polskich znaków i spacji. Wyświetl dane poleceniem View.

```
library(datasets)
```

```
path = "C:\\Users\\petitoff\\Desktop\\repos\\UO\\rok 3\\Wprowadzenie do eksploracji  
danych\\lista3\\" # używając podwójnych ukośników
```

```
setwd(path) ## ustawienie ścieżki
```

```
# Załadowanie danych
```

```
wine <- read.csv('wine\\wine.data', header = FALSE)
```

```
# Oto kod zmieniający nazwy kolumn na nazwy w języku polskim:
```

```
nowe_nazwy <- c(
```

```
  "Klasa",
```

```
  'Alkohol',
```

```
  'Kwas jabłkowy',
```

```
  'Popiół',
```

```
  'Alkalność popiołu',
```

'Magnez',
'Całkowite fenole',
'Flawonoidy',
'Fenole nietrwałe',
'Proantocyjanidy',
'Intensywność koloru',
'Odcień',
'Stężenie odwiedlane win',
'Prolina'
)

The screenshot shows the RStudio interface with a dataset named 'wine' loaded. The dataset has 178 observations and 14 variables. The variables are: 'Klasa', 'Alkohol', 'Kwas jabłkowy', 'Pociół', 'Alkalność popi...', 'Magnez', 'Całkowite fenole', 'Flawonoidy', 'Fenole nietrwałe', 'Proantocyjanidy', 'Intensywność koloru', 'Odcień', 'Prolina', and 'Stężenie odwiedlane win'. The console shows the command 'names(wine) <- nowe_nazwy' and 'view(wine)'. The environment pane shows the 'wine' object with 178 observations and 14 variables.

Klasa	Alkohol	Kwas jabłkowy	Pociół	Alkalność popi...	Magnez	Całkowite fenole	Flawonoidy	Fenole nietrwałe	Proantocyjanidy	Intensywność koloru	Odcień
1	14.23	1.71	2.41	15.4	127	2.80	3.06	0.30	2.29	5.680000	
2	13.20	1.78	2.34	11.2	100	2.61	2.76	0.26	1.28	4.380000	
3	13.16	2.26	2.67	18.6	101	2.80	3.24	0.30	2.81	5.680000	
4	14.37	1.85	2.50	16.8	113	1.85	3.49	0.24	2.18	7.880000	
5	13.24	2.39	2.87	21.0	118	2.80	2.89	0.38	1.42	4.520000	
6	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.750000	
7	14.39	1.87	2.45	14.6	96	2.50	2.52	0.30	1.98	5.250000	
8	14.06	2.15	2.61	17.8	121	2.60	2.51	0.31	1.25	5.050000	
9	14.83	1.64	2.17	14.0	97	2.80	2.86	0.29	1.86	5.200000	
10	13.86	1.35	2.27	16.0	98	2.98	3.15	0.22	1.85	7.220000	
11	14.10	2.16	2.30	18.0	105	2.95	3.32	0.22	2.38	5.750000	
12	14.12	1.48	2.32	16.8	95	2.20	2.43	0.26	1.57	5.000000	
13	13.75	1.73	2.41	16.0	89	2.60	2.76	0.29	1.81	5.600000	
14	14.75	1.73	2.39	11.4	91	3.10	3.69	0.43	2.81	5.400000	
15	14.38	1.87	2.38	12.0	102	3.30	3.64	0.29	2.96	7.300000	
16	13.63	1.81	2.70	17.2	112	2.85	2.91	0.30	1.46	7.300000	
17	14.30	1.82	2.72	20.8	120	2.80	3.14	0.33	1.97	6.200000	
18	13.83	1.57	2.62	20.8	115	2.95	3.40	0.40	1.72	6.600000	
19	14.19	1.59	2.46	16.5	108	3.30	3.89	0.32	1.26	6.700000	
20	13.64	3.18	2.56	16.2	116	2.70	3.85	0.17	1.86	6.180000	
21	14.06	1.63	2.38	16.0	126	3.00	3.17	0.24	2.10	5.650000	
22	12.83	3.80	2.65	18.6	102	2.41	2.41	0.25	1.98	4.580000	
23	13.71	1.86	2.36	18.6	101	2.61	2.88	0.27	1.49	3.800000	
24	12.85	1.80	2.52	17.8	95	2.48	2.37	0.28	1.46	3.930000	
25	13.50	1.81	2.61	20.0	96	2.53	2.61	0.28	1.46	3.520000	
26	13.05	2.05	3.22	25.0	124	2.63	2.88	0.47	1.82	3.580000	

c) Zmienne o wartościach logicznych (jeżeli są) zapisz jako logiczne (polecenie as.logical).

Environment: 178 obs. of 14 variables

values

brakujące log [1:178] FALSE FALSE FALSE FALSE FALSE ...

kolumna chr [1:13] "Alkohol" "Kwas jabłkowy" "Popiół" "Alkalność popi..."

nowe_nazwy chr [1:14] "Klasa" "Alkohol" "Kwas jabłkowy" "Popiół" "Alkaln..."

path chr [1:13] "C:\\Users\\petitoff\\Desktop\\wpos\\u0\\vrok 3\\wprowadzenie..."

wprowadzone chr [1:13] "Alkohol" "Kwas jabłkowy" "Popiół" "Alkalność popi..."

zmienna chr [1:13] "Prolina"

```

> names(wine) <- nowe_nazwy
> View(wine)
> # C) Oto kod zapisujący zmienne o wartościach logicznych jako logiczne:
> wine$Klasa <- as.logical(wine$Klasa)
> View(wine)
> nowe_nazwy

```

wine\$Klasa <- as.logical(wine\$Klasa)

d) Upewnij się, że zmienne o wartościach liczbowych są typu liczbowego, a jeżeli nie są, to zapisz je jako numeryczne (as.numeric)

wprowadzone <- c('Alkohol', 'Kwas jabłkowy', 'Popiół', 'Alkalność popiołu', 'Magnez', 'Całkowite fenole', 'Flawonoidy', 'Fenole nietrwałe', 'Proantocyjanidy', 'Intensywność koloru', 'Odcień', 'Stężenie odwiedlane win', 'Prolina')

```

for (zmienna in wprowadzone) {
  if (class(wine[[zmienna]]) != "numeric") {
    wine[[zmienna]] <- as.numeric(wine[[zmienna]])
  }
}

```

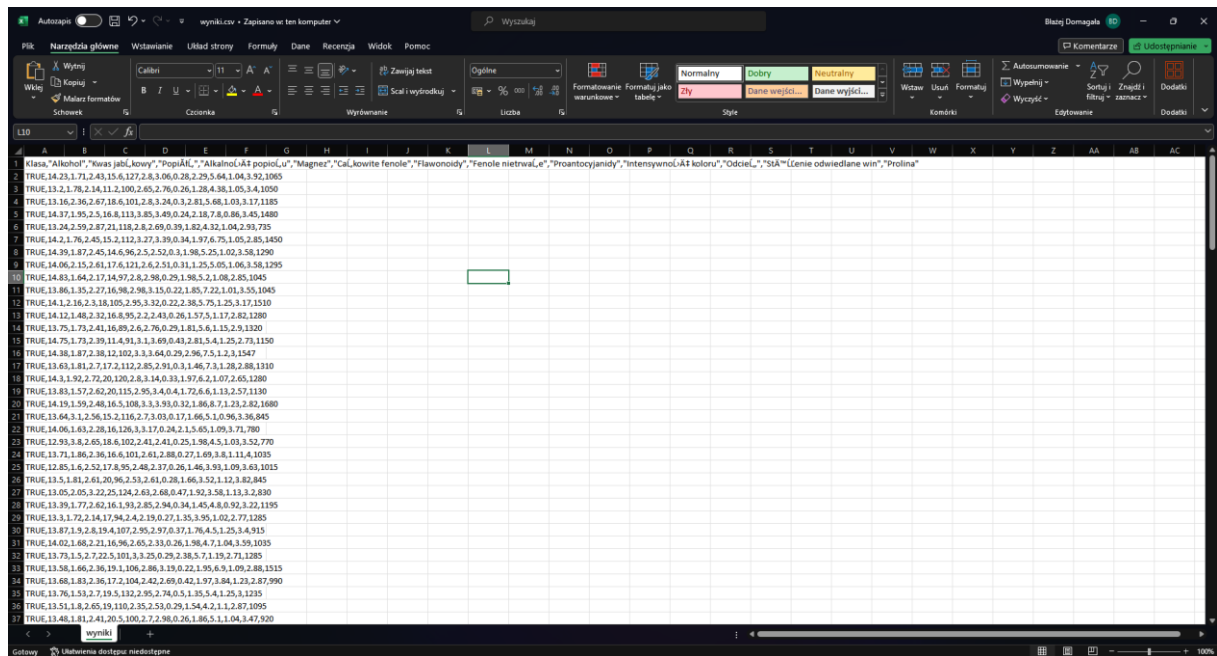
e) Zmienną celu zapisz jako mającą wartości nominalne (polecenie as.factor).

wine\$Klasa <- as.factor(wine\$Klasa)

f) Policz brakujące wartości. Jeżeli są, to dla kolumn o wartościach liczbowych zastąp je wartościami średnimi dla kolumn.

```
for (kolumna in kolumny_numeryczne) {  
  brakujace <- is.na(wine[[kolumna]])  
  if (sum(brakujace) > 0) {  
    srednia <- mean(wine[[kolumna]], na.rm = TRUE)  
    wine[[kolumna]][brakujace] <- srednia  
  }  
}
```

```
write.csv(wine, file = 'wyniki.csv', row.names = FALSE)
```



Lista 4

a) Obliczy i narysuje macierz korelacji zmiennych (bez zmiennej celu wyznaczającej klasy)

```
library(ggplot2)
```

```
library(reshape2)
```

```
correlation_matrix <- function(df, threshold) {
```

```
  # Obliczanie macierzy korelacji
```

```
  cor_matrix <- cor(df)
```

```
  # Rysowanie macierzy korelacji
```

```
  melted_cor_matrix <- melt(cor_matrix)
```

```
  plot <- ggplot(data = melted_cor_matrix, aes(x=Var1, y=Var2, fill=value)) +
```

```
    geom_tile() +
```

```
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
```

```
                        midpoint = 0, limit = c(-1,1), space = "Lab",
```

```
                        name="Pearson\nCorrelation") +
```

```
    theme_minimal() +
```

```
    theme(axis.text.x = element_text(angle = 45, vjust = 1,
```

```
        size = 12, hjust = 1),
```

```
        axis.text.y = element_text(size = 12)) +
```

```
    coord_fixed()
```

```
  # Zapisywanie rysunku do pliku
```

```
  ggsave("correlation_matrix.png", plot)
```

```
  # Wypisywanie par zmiennych o korelacji większej niż zadany próg
```

```
  cor_pairs <- subset(melted_cor_matrix, abs(value) > threshold & Var1 != Var2)
```

```
  # Usuwanie powtórzeń
```



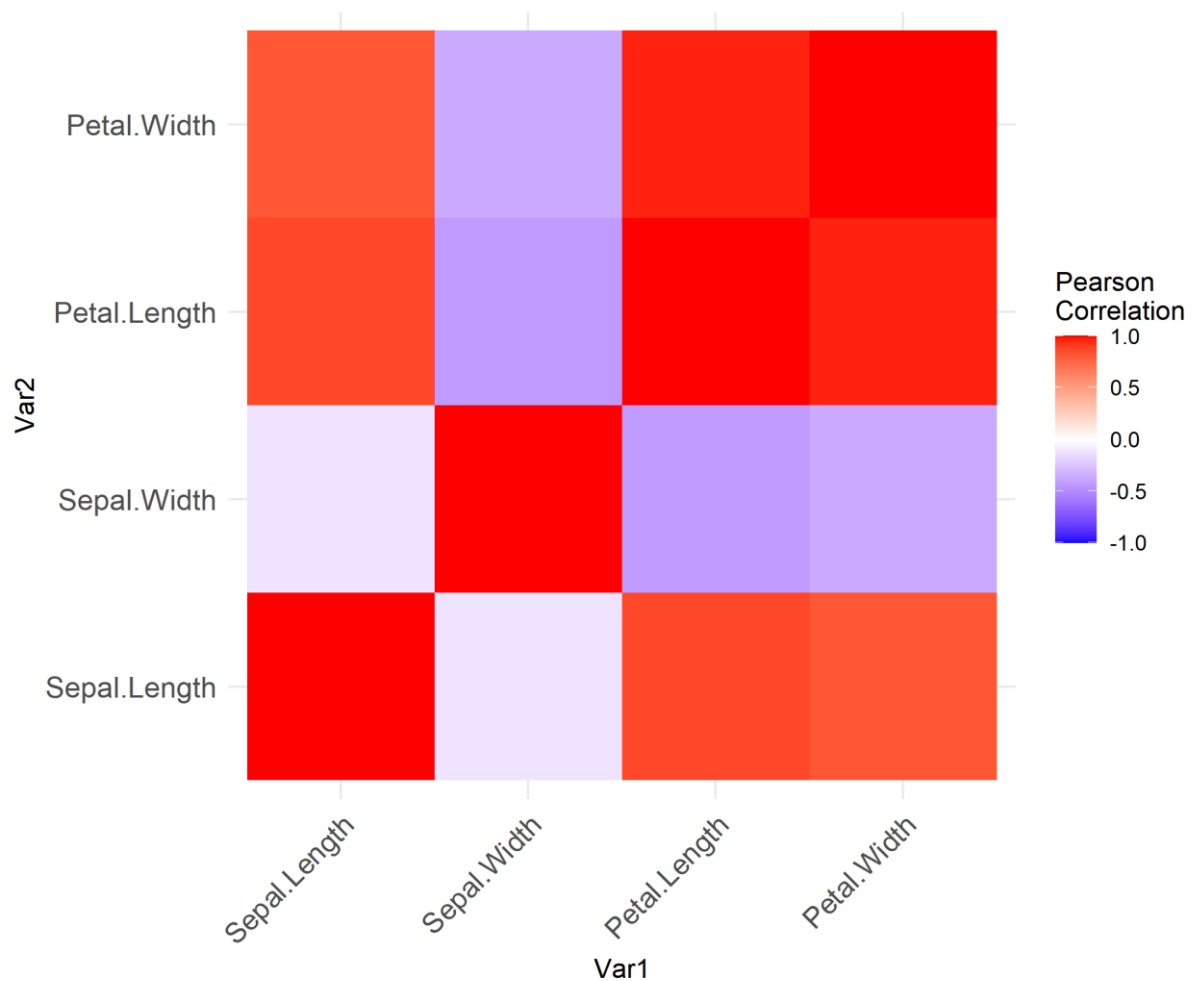
```
cor_pairs <- cor_pairs[!duplicated(t(apply(cor_pairs[,c("Var1", "Var2")], 1, sort))),]  
  
return(cor_pairs)  
}
```

```
# Załadowanie zestawu danych iris  
data(iris)
```

```
# Usunięcie kolumny Species (bo to jest nasza zmienna celu)  
df <- iris[,-5]
```

```
# Użycie funkcji na df z progiem 0.5  
correlation_matrix(df, 0.5)
```

b) zapisze rysunek macierzy korelacji do pliku



c) wypisze pary (nazwy) zmiennych o korelacji większej niż zadany próg oraz odpowiadające im wartości korelacji (wartość progu powinna być argumentem funkcji).

- Proszę uwzględnić ujemne wartości korelacji; czyli przyjmujemy, że np. korelacja równa -0.95 jest powyżej progu 0.9, bo jest to silna korelacja, tylko ujemna (wraz ze wzrostem wartości jednej cechy następuje spadek wartości drugiej cechy).
- Pary proszę wypisać bez powtórek (czyli jeżeli mamy już korelację cechy x z cechą y, to nie wypisujemy korelacji cechy y z x).

Wyniki z konsoli:

```
Saving 7 x 7 in image
      Var1      Var2    value
3 Petal.Length Sepal.Length 0.8717538
4  Petal.Width Sepal.Length 0.8179411
12 Petal.Width Petal.Length 0.9628654
>
```

Lista 5

Dokumentacja Kodu R- Zadanie 2

```
# Biblioteka datasets
library(datasets)

# Ustawienie ścieżki dostępu do danych
path = "C:\\Users\\petit\\Desktop\\repos\\UO\\rok 3\\Wprowadzenie do
eksploracji danych\\lista5\\"
setwd(path)

# ----- Zadanie 2a: Wczytanie i identyfikacja punktów
oddalonych ----- #
# Wczytanie danych
wine_data <- read.csv('wine\\wine.data', header = FALSE)

# Funkcja do identyfikacji i zliczania punktów oddalonych przy
użyciu IQR
identify_outliers <- function(data) {
  quantiles <- quantile(data, c(.25, .75), na.rm = TRUE)
  iqr <- IQR(data, na.rm = TRUE)
  lower_bound <- quantiles[1] - 1.5 * iqr
  upper_bound <- quantiles[2] + 1.5 * iqr
  outliers <- data[data < lower_bound | data > upper_bound]
  return(list("dolna_granica" = lower_bound, "gorna_granica" =
upper_bound, "punkty_oddalone" = outliers))
}

# Część a: Wykrywanie punktów oddalonych dla każdej zmiennej
outliers_info <- lapply(wine_data, identify_outliers)

# Dodanie sztucznego punktu oddalonego, jeśli nie istnieje
for (i in 1:length(outliers_info)) {
```

```

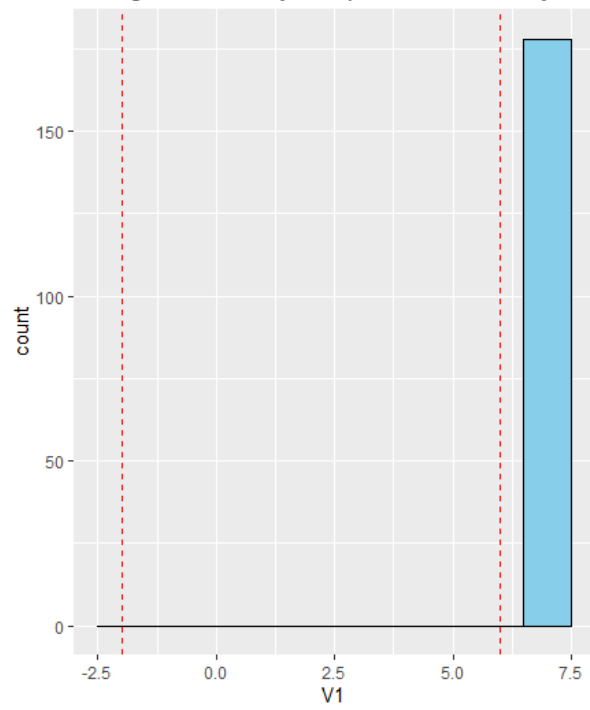
    if (length(outliers_info[[i]]$punkty_oddalone) == 0) {
      wine_data[i][1] <- outliers_info[[i]]$gorna_granica + 1 #
Dodanie punktu oddalonego
    }
  }

# Część b: Wizualizacja punktów oddalonych dla maksymalnie 4
zmiennych

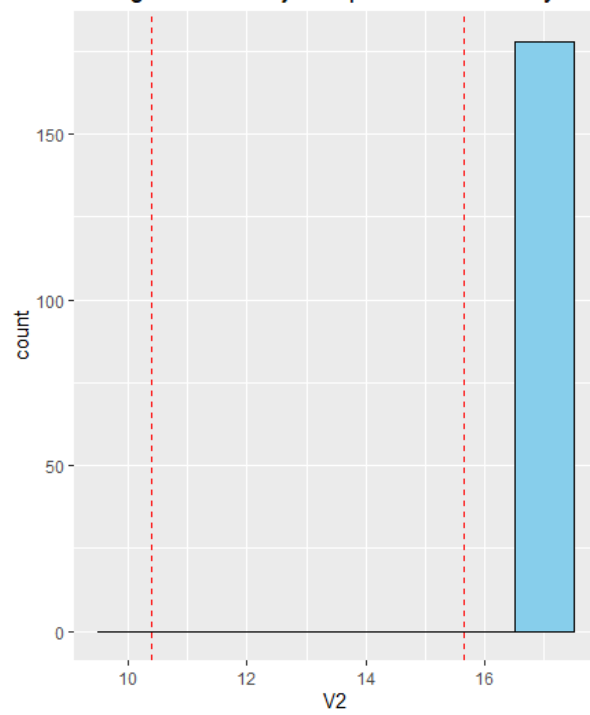
# Wybór pierwszych 4 zmiennych do wizualizacji
selected_vars <- head(names(wine_data), 4)
plots <- list()
for (var in selected_vars) {
  plot <- ggplot(wine_data, aes_string(x=var)) +
    geom_histogram(binwidth = 1, fill="skyblue", color="black") +
    geom_vline(xintercept = outliers_info[[var]]$dolna_granica,
color="red", linetype="dashed") +
    geom_vline(xintercept = outliers_info[[var]]$gorna_granica,
color="red", linetype="dashed") +
    ggtitle(paste("Histogram zmiennej", var, "z punktami
oddalonymi"))
  plots[[var]] <- plot
}

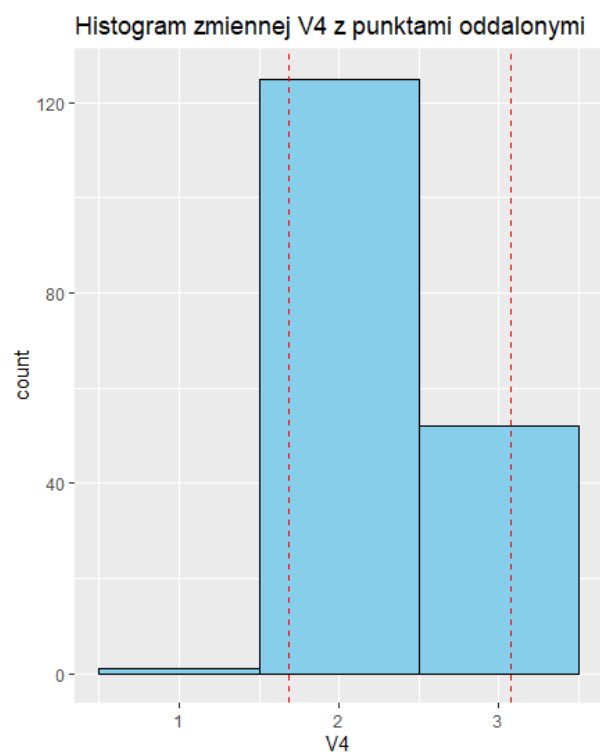
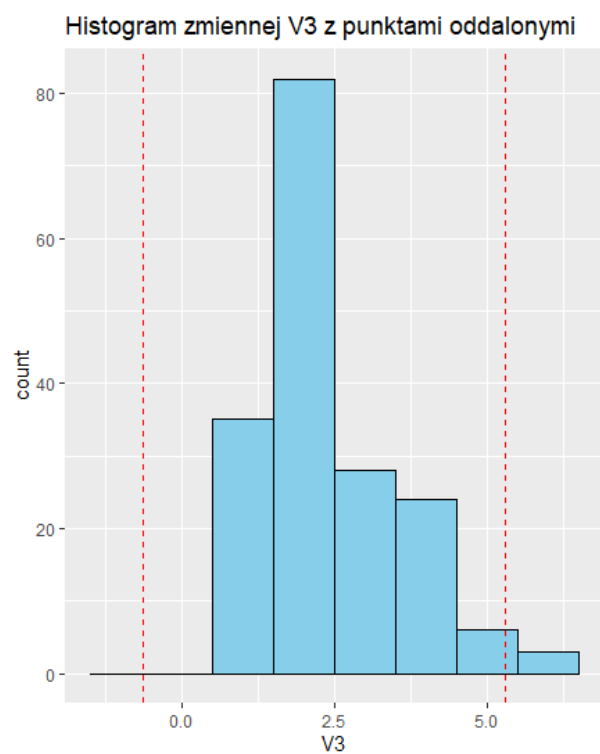
```

Histogram zmiennej V1 z punktami oddalonymi



Histogram zmiennej V2 z punktami oddalonymi





Część c: Usuwanie punktów oddalonych

```
cleaned_wine_data <- wine_data
```

```
for (var in names(wine_data)) {
```

```
  cleaned_wine_data <- cleaned_wine_data[cleaned_wine_data[[var]] >=  
  outliers_info[[var]]$dolna_granica &
```

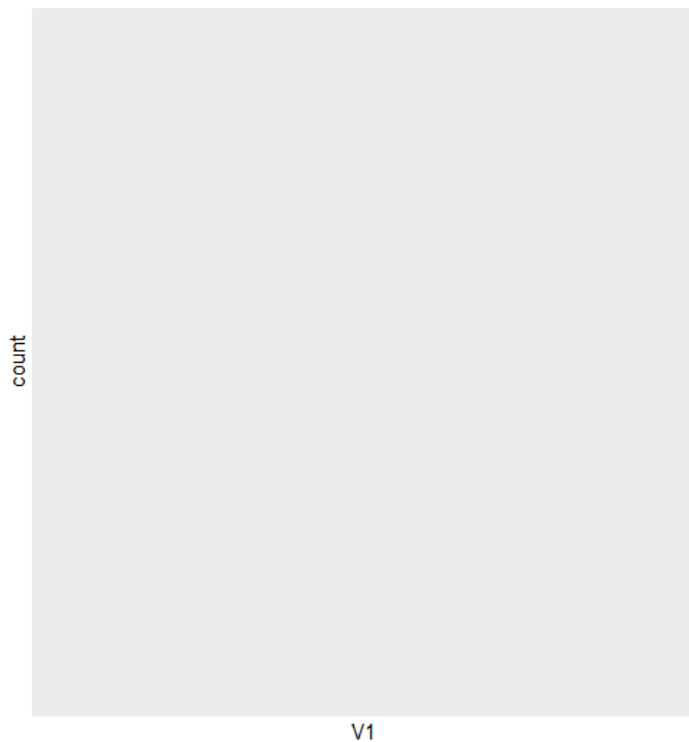
```

                                cleaned_wine_data[[var]]
<= outliers_info[[var]]$gorna_granica, ]
}

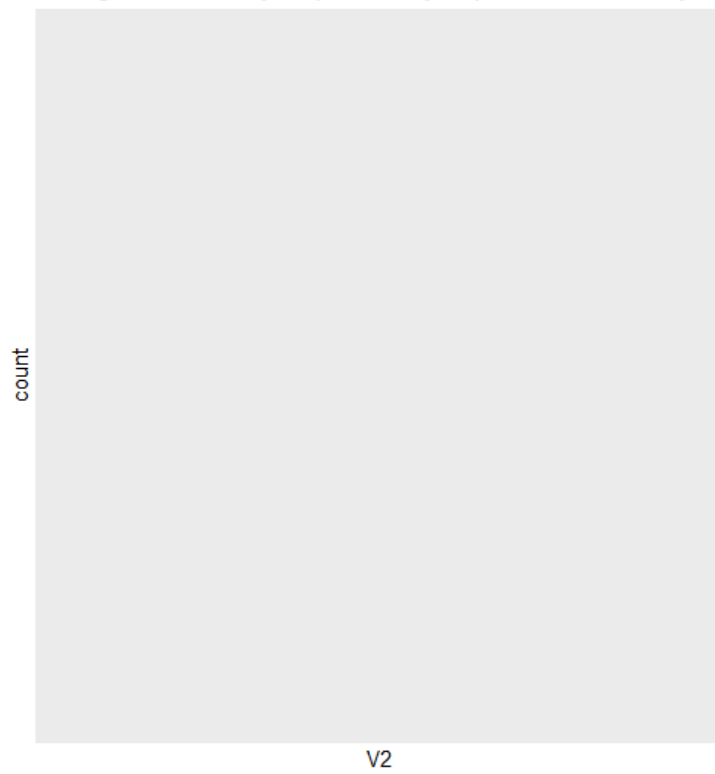
# Część d: Wizualizacja po usunięciu punktów oddalonych dla tych
samyh 4 zmiennych
cleaned_plots <- list()
for (var in selected_vars) {
  plot <- ggplot(cleaned_wine_data, aes_string(x=var)) +
    geom_histogram(binwidth = 1, fill="green", color="black") +
    ggtitle(paste("Histogram zmiennej", var, "po usunięciu punktów
oddalonych"))
  cleaned_plots[[var]] <- plot
}

```

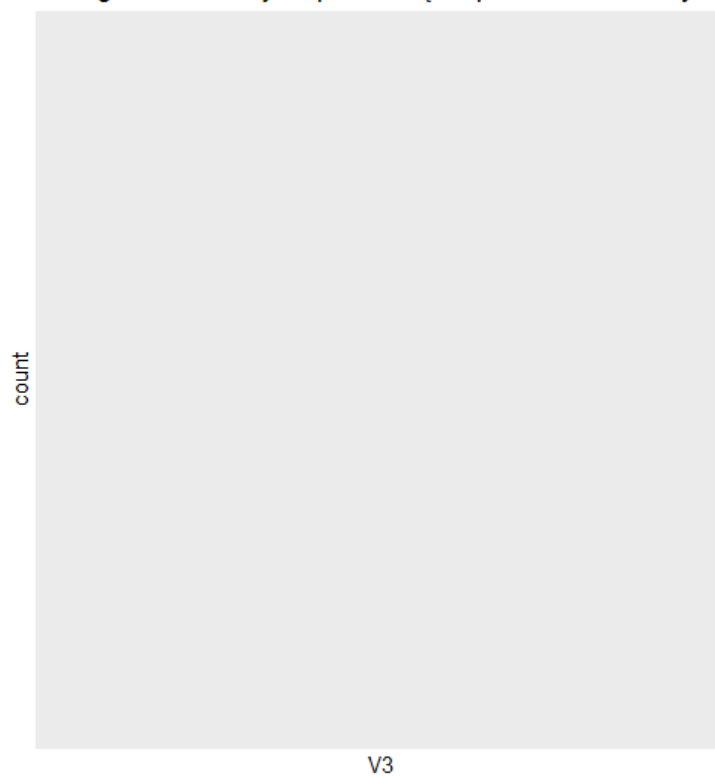
Histogram zmiennej V1 po usunięciu punktów oddalonych



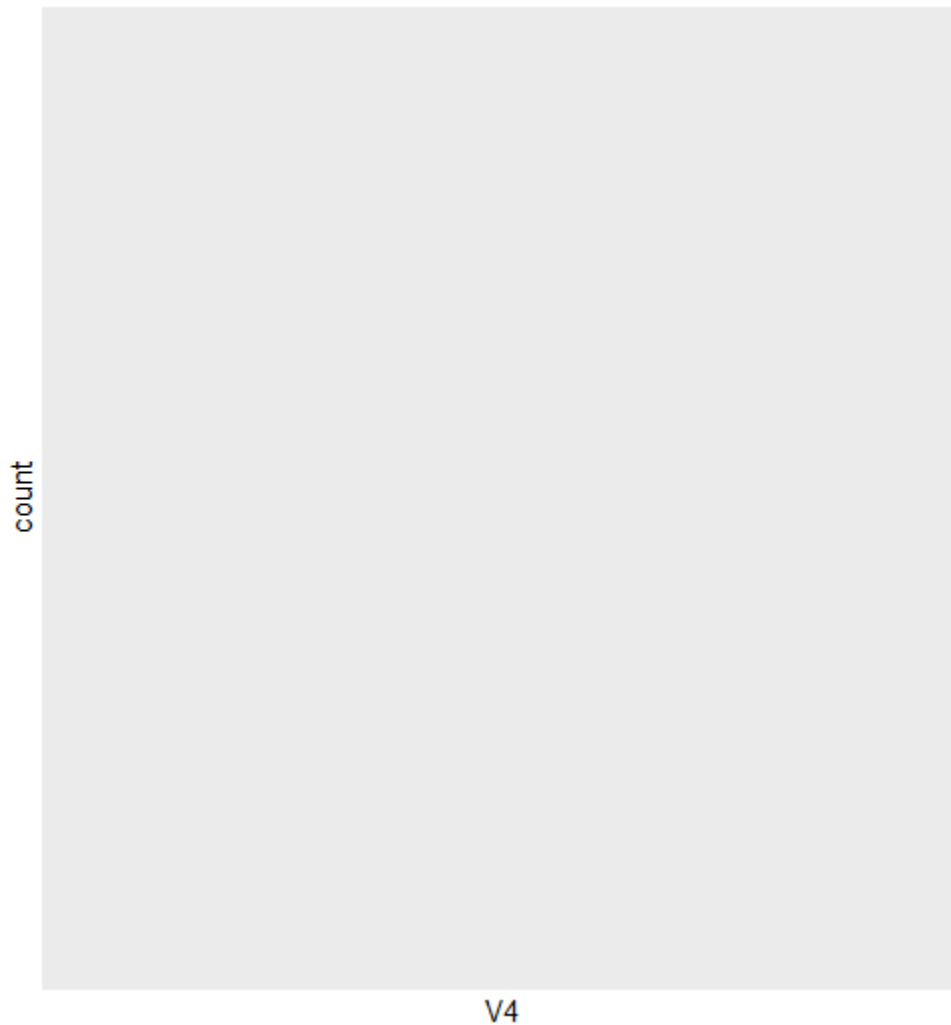
Histogram zmiennej V2 po usunięciu punktów oddalonych



Histogram zmiennej V3 po usunięciu punktów oddalonych



Histogram zmiennej V4 po usunięciu punktów oddalonych



```
# Część e: Zapisanie oczyszczonego zbioru danych
write.csv(cleaned_wine_data, "cleaned_wine_data.csv", row.names =
FALSE)

# Wynik
list("oryginalne_wykresy" = plots, "oczyszczone_wykresy" =
cleaned_plots, "plik_z_oczyszczonymi_danymi" =
"cleaned_wine_data.csv")
```

Lista 6

```
# Wczytanie danych

wine <- read.csv('wine\\wine.data', header = FALSE)

wine_features <- wine[, -1]

# utworzenie macierzy korelacji

cor_matrix <- cor(wine_features)

print(cor_matrix)
```

```
      V2      V3      V4      V5      V6      V7      V8      V9
V2  1.0000000 0.09439694 0.211544596 -0.31023514 0.27079823 0.28910112 0.2368149 -0.1559295
V3  0.09439694 1.00000000 0.164045470 0.28850040 -0.05457510 -0.33516700 -0.4110066 0.2929771
V4  0.21154460 0.16404547 1.000000000 0.44336719 0.28658669 0.12897954 0.1150773 0.1862304
V5 -0.31023514 0.28850040 0.443367187 1.00000000 -0.08333309 -0.32111332 -0.3513699 0.3619217
V6  0.27079823 -0.05457510 0.286586691 -0.08333309 1.00000000 0.21440123 0.1957838 -0.2562940
V7  0.28910112 -0.33516700 0.128979538 -0.32111332 0.21440123 1.00000000 0.8645635 -0.4499353
V8  0.23681493 -0.41100659 0.115077279 -0.35136986 0.19578377 0.86456350 1.0000000 -0.5378996
V9 -0.15592947 0.29297713 0.186230446 0.36192172 -0.25629405 -0.44993530 -0.5378996 1.0000000
V10 0.13669791 -0.22074619 0.009651935 -0.19732684 0.23644061 0.61241308 0.6526918 -0.3658451
V11 0.54636420 0.24898534 0.258887259 0.01873198 0.19995001 -0.05513642 -0.1723794 0.1390570
V12 -0.07174720 -0.56129569 -0.074666889 -0.27395522 0.05539820 0.43368134 0.5434786 -0.2626396
V13 0.07234319 -0.36871043 0.003911231 -0.27676855 0.06600394 0.69994936 0.7871939 -0.5032696
V14 0.64372004 -0.19201056 0.223626264 -0.44059693 0.39335085 0.49811488 0.4941931 -0.3113852

      V10      V11      V12      V13      V14
V2  0.136697912 0.54636420 -0.07174720 0.072343187 0.6437200
V3 -0.220746187 0.24898534 -0.56129569 -0.368710428 -0.1920106
V4  0.009651935 0.25888726 -0.07466689 0.003911231 0.2236263
V5 -0.197326836 0.01873198 -0.27395522 -0.276768549 -0.4405969
V6  0.236440610 0.19995001 0.05539820 0.066003936 0.3933508
V7  0.612413084 -0.05513642 0.43368134 0.699949365 0.4981149
V8  0.652691769 -0.17237940 0.54347857 0.787193902 0.4941931
V9 -0.365845099 0.13905701 -0.26263963 -0.503269596 -0.3113852
V10 1.000000000 -0.02524993 0.29554425 0.519067096 0.3304167
V11 -0.025249931 1.00000000 -0.52181319 -0.428814942 0.3161001
V12 0.295544253 -0.52181319 1.00000000 0.565468293 0.2361834
V13 0.519067096 -0.42881494 0.56546829 1.00000000 0.3127611
V14 0.330416700 0.31610011 0.23618345 0.312761075 1.0000000
```

```
# Obliczanie i wydrukowanie indeksów oraz rang cech:

pearson_corr <- sapply(wine_features, function(x) cor(x, wine[, 1]))

pearson_rank <- order(-abs(pearson_corr))

print(pearson_corr)

print(pearson_rank)
```

```

> # obliczanie i wydrukowanie indeksów oraz rang cech:
> pearson_corr <- sapply(wine_features, function(x) cor(x, wine[, 1]))
> pearson_rank <- order(-abs(pearson_corr))
> print(pearson_corr)
      v2      v3      v4      v5      v6      v7      v8      v9
-0.32822194  0.43777620 -0.04964322  0.51785911 -0.20917939 -0.71916334 -0.84749754  0.48910916
      v10      v11      v12      v13      v14
-0.49912982  0.26566757 -0.61736921 -0.78822959 -0.63371678
> print(pearson_rank)
[1] 7 12 6 13 11 4 9 8 2 1 10 5 3
>

```

Utworzenie tablicy 2D z rangami:

```

ranks_matrix <- data.frame(pearson = pearson_rank)
rownames(ranks_matrix) <- colnames(wine_features)
print(ranks_matrix)

```

```

# print(ranks_matrix)
      pearson
v2          7
v3         12
v4          6
v5         13
v6         11
v7          4
v8          9
v9          8
v10         2
v11         1
v12        10
v13         5
v14         3

```

Dodanie kolumny ze średnią wartością rangi:

```

ranks_matrix$average_rank <- rowMeans(ranks_matrix, na.rm = TRUE)
print(ranks_matrix)

```

```
> print(ranks_matrix)
      pearson average_rank
v2          7           7
v3         12          12
v4          6           6
v5         13          13
v6         11          11
v7          4           4
v8          9           9
v9          8           8
v10         2           2
v11         1           1
v12        10          10
v13         5           5
v14         3           3
```

Sortowanie cech według wartości średniej rangi:

```
sorted_ranks <- ranks_matrix[order(ranks_matrix$average_rank), ]
print(sorted_ranks)
```

```
> print(sorted_ranks)
      pearson average_rank
v11         1           1
v10         2           2
v14         3           3
v7          4           4
v13         5           5
v4          6           6
v2          7           7
v9          8           8
v8          9           9
v12        10          10
v6         11          11
v3         12          12
v5         13          13
```

Spearman Correlation

```
spearman_corr <- sapply(wine_features, function(x) cor(x, wine[, 1],
method = "spearman"))

spearman_rank <- order(-abs(spearman_corr))

ranks_matrix$spearman <- spearman_rank

print(spearman_corr)

print(spearman_rank)
```

```

> # Spearman Correlation
> spearman_corr <- sapply(wine_features, function(x) cor(x, wine[, 1], method = "spearman"))
> spearman_rank <- order(-abs(spearman_corr))
> ranks_matrix$spearman <- spearman_rank
> print(spearman_corr)
      V2      V3      V4      V5      V6      V7
-0.35416692 0.34691327 -0.05398792 0.56979214 -0.25049819 -0.72654365
      V8      V9      V10     V11     V12     V13
-0.85490766 0.47420549 -0.57064758 0.13117017 -0.61657049 -0.74378690
      V14
-0.57638313
> print(spearman_rank)
[1] 7 12 6 11 13 9 4 8 1 2 5 10 3

```

Kendall Correlation

```

kendall_corr <- sapply(wine_features, function(x) cor(x, wine[, 1],
method = "kendall"))

```

```

kendall_rank <- order(-abs(kendall_corr))

```

```

ranks_matrix$kendall <- kendall_rank

```

```

print(kendall_corr)

```

```

print(kendall_rank)

```

```

> # Kendall Correlation
> kendall_corr <- sapply(wine_features, function(x) cor(x, wine[, 1], method = "kendall"))
> kendall_rank <- order(-abs(kendall_corr))
> ranks_matrix$kendall <- kendall_rank
> print(kendall_corr)
      V2      V3      V4      V5      V6      V7
-0.23898423 0.24749447 -0.03808511 0.44940228 -0.18499225 -0.59040381
      V8      V9      V10     V11     V12     V13
-0.72525486 0.37923359 -0.45022461 0.06512382 -0.47922876 -0.60757229
      V14
-0.40626000
> print(kendall_rank)
[1] 7 12 6 11 9 4 13 8 2 1 5 10 3
>

```

Mutual Information

```

mi_scores <- sapply(wine_features, function(x)
mutinformation(discretize(x), discretize(wine[, 1])))

```

```

mi_rank <- order(-mi_scores)

```

```

ranks_matrix$MI <- mi_rank

```

```

print(mi_scores)

```

```

print(mi_rank)

```

```

> # Mutual Information
> mi_scores <- sapply(wine_features, function(x) mutinformation(discretize(x), discretize(wine[, 1])))
> mi_rank <- order(-mi_scores)
> ranks_matrix$MI <- mi_rank
> print(mi_scores)
      V2      V3      V4      V5      V6      V7      V8
0.43718816 0.26059723 0.09409405 0.19209690 0.19419468 0.36548167 0.61456543
      V9      V10     V11     V12     V13     V14
0.17079017 0.21654374 0.47156977 0.40134555 0.48896304 0.53321967
> print(mi_rank)
[1] 7 13 12 10 1 11 6 2 9 5 4 8 3

```

```
# PCA - pierwsza główna składowa
pca_result <- prcomp(wine_features, scale. = TRUE)
pca_rank <- order(-abs(pca_result$rotation[, 1]))
ranks_matrix$PCA <- pca_rank
print(pca_result$rotation[, 1])
print(pca_rank)
```

```
>
> # PCA - pierwsza główna składowa
> pca_result <- prcomp(wine_features, scale. = TRUE)
> pca_rank <- order(-abs(pca_result$rotation[, 1]))
> ranks_matrix$PCA <- pca_rank
> print(pca_result$rotation[, 1])
      V2      V3      V4      V5      V6      V7
-0.144329395 0.245187580 0.002051061 0.239320405 -0.141992042 -0.394660845
      V8      V9     V10     V11     V12     V13
-0.422934297 0.298533103 -0.313429488 0.088616705 -0.296714564 -0.376167411
      V14
-0.286752227
> print(pca_rank)
[1] 7 6 12 9 8 11 13 2 4 1 5 10 3
```

```
# LDA
lda_result <- lda(wine[, 1] ~ ., data = wine_features)
lda_rank <- order(-abs(lda_result$scaling[, 1]))
ranks_matrix$LDA <- lda_rank
print(lda_result$scaling[, 1])
print(lda_rank)
```

```
>
> # LDA
> lda_result <- lda(wine[, 1] ~ ., data = wine_features)
> lda_rank <- order(-abs(lda_result$scaling[, 1]))
> ranks_matrix$LDA <- lda_rank
> print(lda_result$scaling[, 1])
      V2      V3      V4      V5      V6      V7
-0.403399781 0.165254596 -0.369075256 0.154797889 -0.002163496 0.618052068
      V8      V9     V10     V11     V12     V13
-1.661191235 -1.495818440 0.134092628 0.355055710 -0.818036073 -1.157559376
      V14
-0.002691206
> print(lda_rank)
[1] 7 8 12 11 6 1 3 10 2 4 9 13 5
```

```
# Chi-Square Test
```

```
# Uwaga: Ten test ma sens tylko dla cech kategoryalnych.
```

```
# Aktualizacja średniej rangi
```

```
ranks_matrix$average_rank <- rowMeans(ranks_matrix[, -1], na.rm = TRUE)
```

```
print(ranks_matrix$average_rank)
```

```
>
> # Chi-Square Test
> # Uwaga: Ten test ma sens tylko dla cech kategoryalnych.
>
> # Aktualizacja średniej rangi
> ranks_matrix$average_rank <- rowMeans(ranks_matrix[, -1], na.rm = TRUE)
> print(ranks_matrix$average_rank)
[1] 7.000000 10.500000 9.000000 10.833333 8.000000 6.666667 8.000000 6.333333 3.333333
[10] 2.333333 6.333333 9.333333 3.333333
>
> # Sortowanie cech według wartości średniej rangi
```

```
# Sortowanie cech według wartości średniej rangi
```

```
sorted_ranks <- ranks_matrix[order(ranks_matrix$average_rank), ]
print(sorted_ranks)
```

```
>
> # Sortowanie cech według wartości średniej rangi
> sorted_ranks <- ranks_matrix[order(ranks_matrix$average_rank), ]
> print(sorted_ranks)
      pearson average_rank spearman kendall MI PCA LDA
V11         1      2.333333         2         1  5  1  4
V10         2      3.333333         1         2  9  4  2
V14         3      3.333333         3         3  3  3  5
V9          8      6.333333         8         8  2  2 10
V12        10      6.333333         5         5  4  5  9
V7          4      6.666667         9         4 11 11  1
V2          7      7.000000         7         7  7  7  7
V6         11      8.000000        13         9  1  8  6
V8          9      8.000000         4        13  6 13  3
V4          6      9.000000         6         6 12 12 12
V13         5      9.333333        10        10  8 10 13
V3         12     10.500000        12        12 13  6  8
V5         13     10.833333        11        11 10  9 11
> |
```

Wnioski

1. Ważność Cech wg Korelacji Pearsona:

- Cecha **V11** wykazuje najwyższą korelację (najniższą rangę) z etykietami klas, co wskazuje na jej potencjalnie dużą wagę w modelowaniu.
- Z kolei cecha **V5** ma najniższą korelację (najwyższą rangę), co sugeruje jej mniejsze znaczenie.

2. Różnorodność Korelacji:

- Obserwujemy różnorodne wartości korelacji, od silnie negatywnych (np. **V7**, **V8**, **V13**) do silnie pozytywnych (np. **V4**, **V10**, **V11**), co wskazuje na złożoność zależności w tych danych.

3. Znaczenie dla Modelowania:

- Cechy o wyższej korelacji mogą być bardziej znaczące w modelach predykcyjnych, jednak warto pamiętać, że korelacja nie zawsze równa się przyczynowości.
- Cechy o niskiej korelacji nie powinny być automatycznie odrzucane, gdyż mogą wnosić cenne informacje w połączeniu z innymi cechami.

4. Wnioski z Innych Metod Filtracyjnych:

- **Korelacje Spearmana i Kendalla:** Te metody wskazują na monotoniczne związki między cechami a etykietami klas. Zauważalna jest pewna konsystencja z wynikami korelacji Pearsona, co dodatkowo potwierdza ważność niektórych cech (np. V11, V10).
- **Wzajemna Informacja (MI):** Ta metoda ocenia wzajemną zależność między zmiennymi, co może ujawnić nieliniowe związki. Cechy z wysokim wynikiem MI mogą odgrywać kluczową rolę w rozróżnianiu klas, nawet jeśli ich liniowa korelacja jest słaba.

5. Analiza Głównych Składowych (PCA):

- Wyniki PCA mogą pomóc zrozumieć, które cechy najbardziej przyczyniają się do wariancji w zbiorze danych. Cechy z wysokimi wartościami na pierwszej głównej składowej mogą być istotne dla różnorodności w danych.

6. Analiza Dyskryminacyjna Liniowa (LDA):

- Wyniki LDA podkreślają cechy, które najlepiej rozróżniają klasy. Ta metoda jest szczególnie przydatna w kontekście klasyfikacji i może wskazywać na cechy krytyczne dla rozróżniania między kategoriami wina.

7. Ważność Integracji Różnych Metod:

- Integracja wyników z różnych metod daje bardziej zrównoważony widok na ważność cech. Cechy, które są konsekwentnie wysoko oceniane przez różne metody, są prawdopodobnie kluczowe dla zrozumienia i modelowania zbioru danych.

8. Praktyczne Implikacje dla Modelowania:

- Cechy o wysokich średnich rangach z różnych metod filtracyjnych mogą być priorytetowe przy tworzeniu modeli predykcyjnych.
- Warto jednak zachować ostrożność i nie wykluczać cech o niższych rangach, ponieważ mogą one odgrywać istotne role w interakcjach z innymi zmiennymi.

Lista 7

1. **Wczytanie danych:** Import danych z pliku **wine.data** bez nagłówek.

```
# Wczytanie danych
```

```
wine_data <- read.csv('wine\\wine.data', header = FALSE)
```

2. **Usunięcie brakujących danych:** Wyeliminowanie obserwacji z brakującymi wartościami.

3. **Usunięcie punktów oddalonych:** Obliczenie odległości Mahalanobisa dla każdego punktu, ustalenie progu (95 percentyl) i usunięcie punktów przekraczających ten próg.

```
# Usunięcie obserwacji brakujących i punktów oddalonych
```

```
wine_data <- na.omit(wine_data) # Usunięcie obserwacji brakujących
```

```
# Obliczanie odległości Mahalanobisa dla każdego punktu
```

```
wine_data$distance <- mahalanobis(wine_data, colMeans(wine_data),  
cov(wine_data))
```

```
# Ustalenie progu dla identyfikacji punktów oddalonych, np. 95  
percentyla
```

```
threshold <- quantile(wine_data$distance, 0.95)
```

```
# Usuwanie punktów oddalonych
```

```
wine_data <- wine_data[wine_data$distance <= threshold, ]
```

```
# Przygotowanie danych do PCA
```

```
wine_data_pca <- PCA(wine_data, graph = FALSE)
```

```
# a) Wartości własne i wyjaśniana wariancja
```

```
eig_val <- get_eigenvalue(wine_data_pca)
```

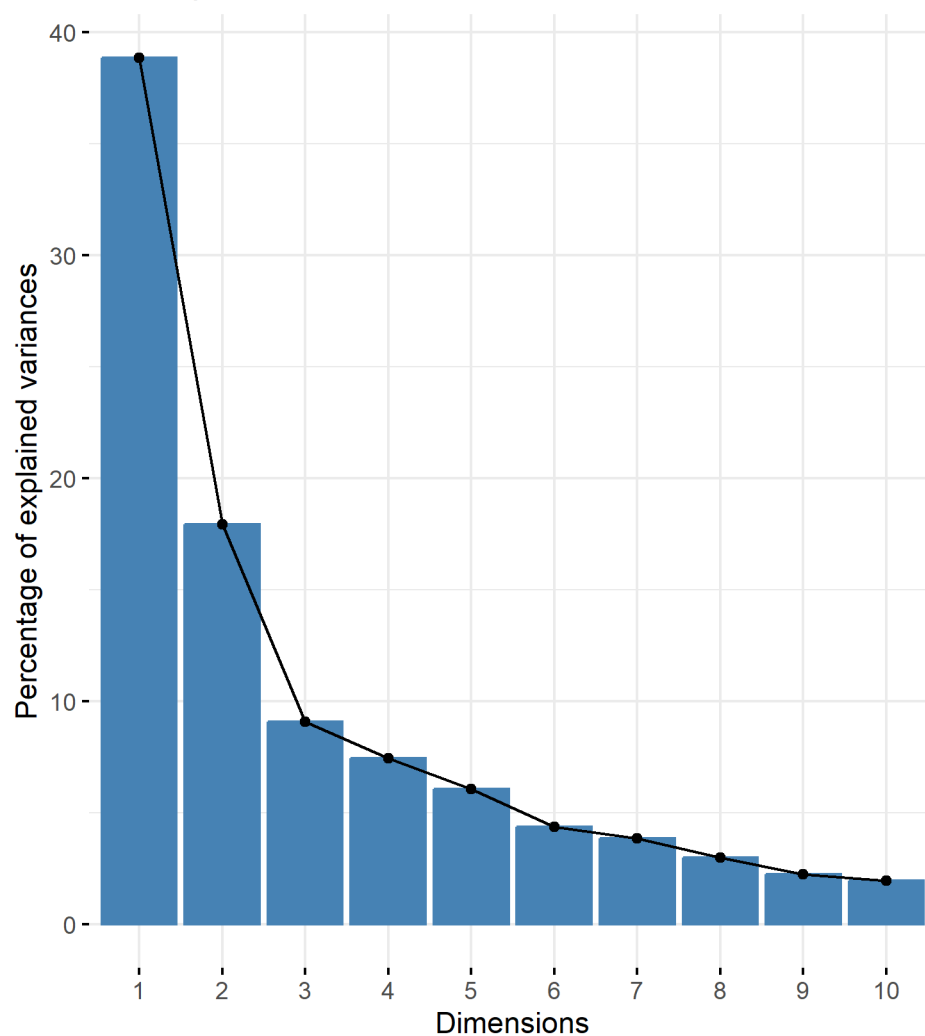
```
print(eig_val)
```

```
fviz_eig(wine_data_pca) # Wykres osypiskowy
```

```
ggsave("scree_plot.png", bg = "white") # Zapisanie do pliku
```

```
> print(eig_val)
      eigenvalue variance.percent cumulative.variance.percent
Dim.1  5.82941960      38.8627973      38.86280
Dim.2  2.69124789      17.9416526      56.80445
Dim.3  1.36266926       9.0844617      65.88891
Dim.4  1.11687523       7.4458349      73.33475
Dim.5  0.91200393       6.0800262      79.41477
Dim.6  0.65538714       4.3692476      83.78402
Dim.7  0.57881234       3.8587489      87.64277
Dim.8  0.45060262       3.0040174      90.64679
Dim.9  0.33766066       2.2510711      92.89786
Dim.10 0.29486852       1.9657902      94.86365
Dim.11 0.24451072       1.6300714      96.49372
Dim.12 0.21036027       1.4024018      97.89612
Dim.13 0.15883684       1.0589123      98.95503
Dim.14 0.11625933       0.7750622      99.73010
Dim.15 0.04048565       0.2699044     100.00000
```

Scree plot



b) Ładunki czynnikowe

```
loadings <- wine_data_pca$var$coord
```

```
print(loadings)
```

```
> # b) Ładunki czynnikowe
> loadings <- wine_data_pca$var$coord
> print(loadings)
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
v1	-0.936928269	0.015146156	-0.01927134	0.18943162	0.02981302
v2	0.367586316	0.769061818	-0.17948661	-0.07777073	0.05030504
v3	-0.522912555	0.370153338	0.07767294	-0.04167927	0.54345198
v4	0.007349719	0.471623556	0.75545529	-0.30885394	-0.07645014
v5	-0.609378200	-0.088850874	0.58443340	-0.20620017	0.18428368
v6	0.296382194	0.621980642	0.21408833	0.15125893	-0.35554663
v7	0.857891058	0.047866699	0.17805513	0.07838012	0.15582113
v8	0.945084062	-0.059908544	0.12677495	0.05406437	0.12959302
v9	-0.650228330	0.006129602	0.16256496	-0.30488966	-0.20176459
v10	0.690515627	-0.014080261	0.20169227	0.25071746	0.35613353
v11	-0.180569880	0.830998790	-0.14142455	0.21855428	0.01484359
v12	0.633475899	-0.456736402	0.14587262	-0.16124894	-0.36342541
v13	0.825279206	-0.293856815	0.08752902	-0.16378616	0.23474509
v14	0.648633381	0.576533095	-0.09169389	-0.13357897	-0.16076558
distance	-0.138966452	-0.146083137	0.44012379	0.79315643	-0.15862579

```
>
```

```
# c) Zasoby zmienności wspólnej
communalities <- wine_data_pca$var$cos2
print(communalities)
```

```
> # c) Zasoby zmienności wspólnej
> communalities <- wine_data_pca$var$cos2
> print(communalities)
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
v1	8.778346e-01	2.294060e-04	0.0003713846	0.035884340	0.0008888159
v2	1.351197e-01	5.914561e-01	0.0322154420	0.006048287	0.0025305968
v3	2.734375e-01	1.370135e-01	0.0060330858	0.001737162	0.2953400600
v4	5.401837e-05	2.224288e-01	0.5707126951	0.095390759	0.0058446242
v5	3.713418e-01	7.894478e-03	0.3415624034	0.042518510	0.0339604752
v6	8.784241e-02	3.868599e-01	0.0458338130	0.022879263	0.1264134034
v7	7.359771e-01	2.291221e-03	0.0317036285	0.006143443	0.0242802256
v8	8.931839e-01	3.589034e-03	0.0160718881	0.002922956	0.0167943500
v9	4.227969e-01	3.757203e-05	0.0264273669	0.092957704	0.0407089506
v10	4.768118e-01	1.982537e-04	0.0406797735	0.062859246	0.1268310942
v11	3.260548e-02	6.905590e-01	0.0200009035	0.047765975	0.0002203321
v12	4.012917e-01	2.086081e-01	0.0212788212	0.026001221	0.1320780319
v13	6.810858e-01	8.635183e-02	0.0076613292	0.026825906	0.0551052595
v14	4.207253e-01	3.323904e-01	0.0084077694	0.017843342	0.0258455728
distance	1.931167e-02	2.134028e-02	0.1937089526	0.629097117	0.0251621400

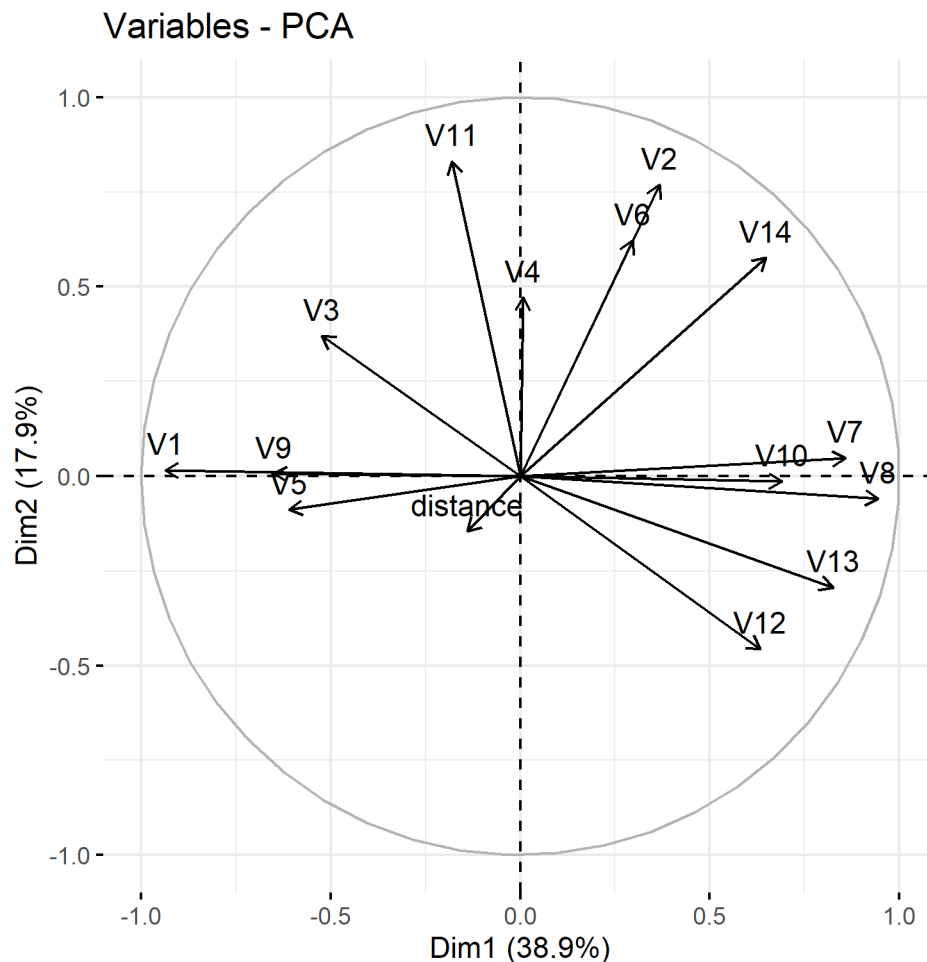
```
# d) Wkłady zmiennych pierwotnych w składowe główne
contributions <- wine_data_pca$var$contrib
print(contributions)

fviz_pca_var(wine_data_pca) # Wykres korelacji
ggsave("correlation_plot.png", bg = "white") # Zapisanie do pliku
```

```
>
> # d) wkłady zmiennych pierwotnych w składowe główne
> contributions <- wine_data_pca$var$contrib
> print(contributions)
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
V1	15.058696078	0.008524151	0.0272542	3.2129229	0.09745746
V2	2.317892847	21.977019751	2.3641424	0.5415365	0.27747652
V3	4.690647762	5.091076681	0.4427403	0.1555377	32.38363888
V4	0.000926651	8.264893769	41.8819675	8.5408608	0.64085516
V5	6.370133147	0.293338934	25.0656865	3.8069168	3.72372026
V6	1.506880805	14.374741194	3.3635317	2.0485066	13.86105904
V7	12.625220323	0.085136003	2.3265828	0.5500564	2.66229396
V8	15.322003641	0.133359458	1.1794416	0.2617084	1.84147780
V9	7.252812631	0.001396082	1.9393823	8.3230159	4.46368148
V10	8.179404876	0.007366610	2.9853006	5.6281351	13.90685827
V11	0.559326381	25.659434546	1.4677739	4.2767512	0.02415911
V12	6.883905134	7.751353650	1.5615544	2.3280328	14.48217790
V13	11.683594835	3.208616641	0.5622296	2.4018713	6.04221731
V14	7.217275341	12.350791308	0.6170073	1.5976128	2.83393217
distance	0.331279548	0.792951222	14.2154049	56.3265349	2.75899468

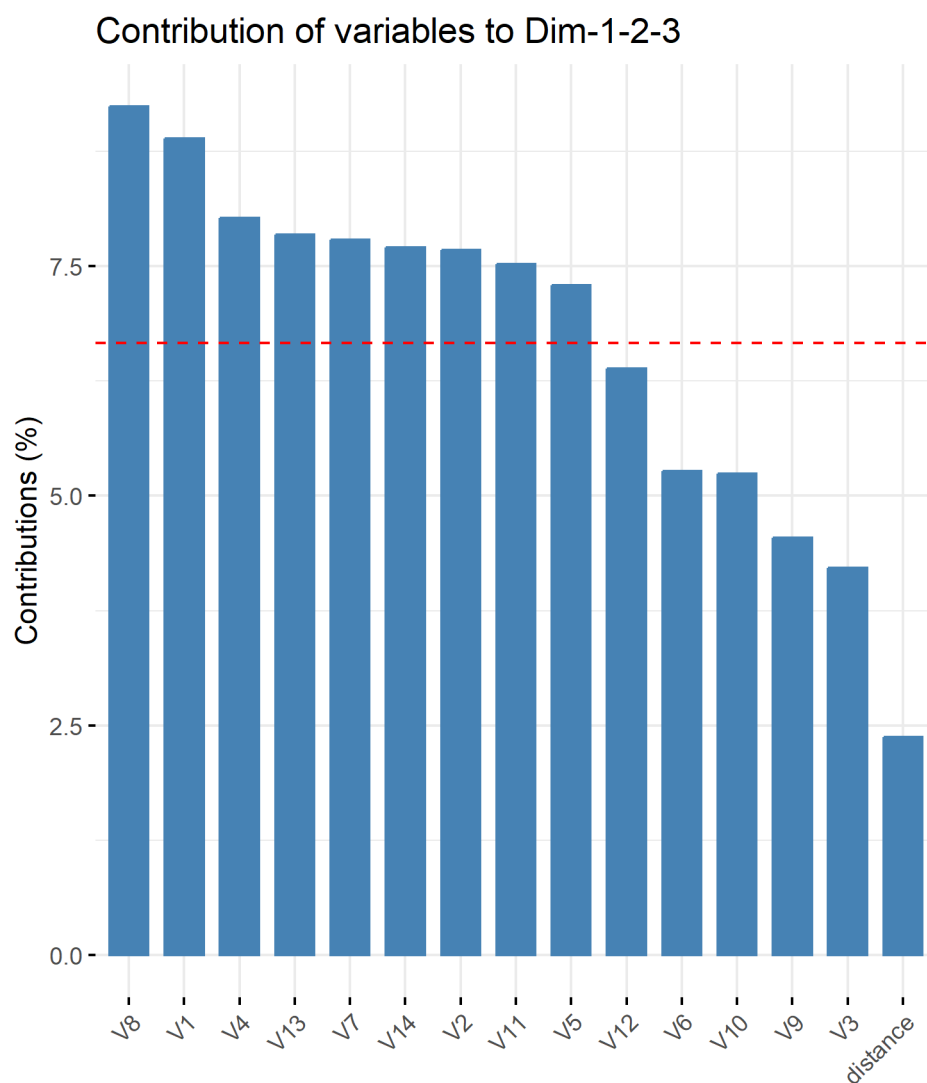
```
> fviz_contrib(wine_data_pca, choice = "var", axes = 1:3)
```



e) Wykres łącznego wkładu dla pierwszych trzech wymiarów

```
fviz_contrib(wine_data_pca, choice = "var", axes = 1:3)
```

```
ggsave("contrib_plot.png", bg = "white")
```



```
# f) Wektory własne
```

```
eigen_vectors <- wine_data_pca$ind$coord
```

```
print(eigen_vectors)
```

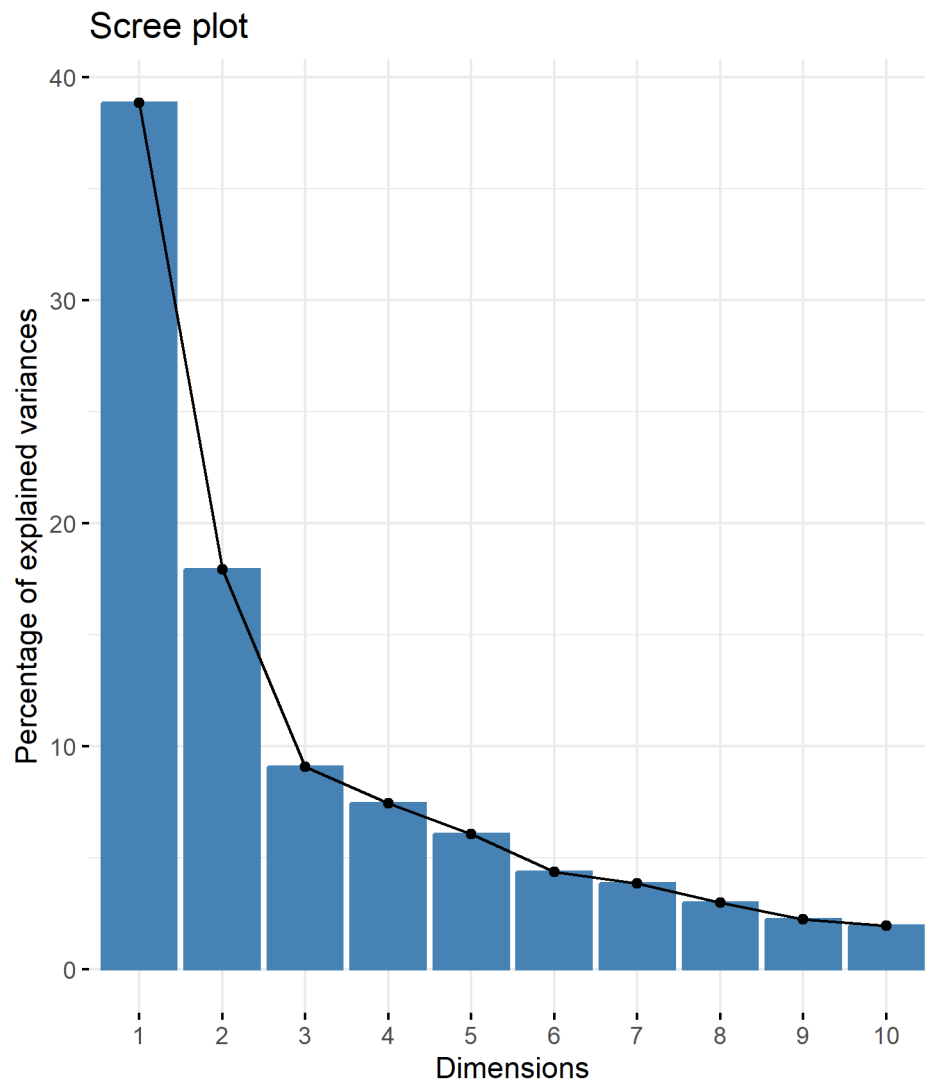
```
# g) Wybór liczby składowych
```

```
# Analiza procentu wyjaśnianej wariancji
```

```
fviz_screplot(wine_data_pca)
```

```
ggsave("explained_variance_plot.png", bg = "white") # Zapisanie do pliku
```

```
# Załóżmy, że decydujemy się zachować pierwsze 3 składowe główne  
selected_components <- 1:3
```



```
# h) Utworzenie nowej tabeli danych
```

```
# Wybór składowych głównych do nowej tabeli danych  
new_data <- wine_data_pca$ind$coord[, selected_components]
```

```
# Konwersja new_data do ramki danych, jeśli to konieczne  
new_data <- as.data.frame(new_data)
```

```
# Dodanie zmiennej celu (klasy) do nowej tabeli  
new_data$class <- wine_data[, 1]
```

```
# Zapis nowej tabeli danych do pliku CSV
if (!require(readr)) install.packages("readr")
library(readr)

write_csv(new_data, "new_wine_data.csv")
```


Lista 8

Standardize the data

```
wine_data_scaled <- scale(wine_data)
```

```
# a) Stosowanie metody k-średnich dla k = 1 do 15
```

```
results <- list()
```

```
for (k in 1:15) {
```

```
  set.seed(123) # dla powtarzalności wyników
```

```
  kmeans_result <- kmeans(wine_data_scaled, centers = k, nstart = 25)
```

```
  results[[k]] <- kmeans_result
```

```
  cat("Dla k =", k, "\n",  
      "Całkowita suma kwadratów wewnątrz klastrów:",  
      kmeans_result$tot.withinss, "\n",  
      "Suma kwadratów dla każdego klastra:", kmeans_result$withinss,  
      "\n",  
      "Centra klastrów:\n", kmeans_result$centers, "\n",  
      "Liczba obserwacji w każdym klastrze:", kmeans_result$size,  
      "\n",  
      "Przypisanie klastrów dla pierwszych 20 obserwacji:",  
      kmeans_result$cluster[1:20], "\n\n")  
}
```

Wyniki

Dołączam plik excel oraz wklejam dla przyszłego wysyłania arkusze z pliku



kmeans_clustering_
updated_results.xls>

Total Within- Cluster Sum of Squares	Within- Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations

-1.227483e-15 -8.719617e-16 -5.738231e-17
8.470128e-16 -1.559302e-16 -6.14365e-17
2.186141e-16 1.135172e-16 6.224734e-16 -
1.503167e-16 2.370139e-17 1.846214e-16
3.492836e-17 -6.112464e-17

[2478]

[178]

[1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
2478	[2478]	1.032546 -0.5939424 -0.07277357 0.0418609 0.6626412 -0.3811653 0.1893414 -0.1089132 0.5151693 -0.2963363 - 0.1542527 0.08872942 -0.9410522 0.5413132 -1.043692 0.6003536 0.835592 - 0.4806503 -0.7141412 0.4107892 0.5419399 -0.3117353 -0.8795908 0.5059593 -1.06631 0.6133643 -0.4519062 0.259946	[178]	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
1717,525	[532.4956, 1185.03]		[65, 113]	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
1030	[100, 101, 102]	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00	[10, 11, 12]	[1, 2, 3, 1, 2, 3]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
1040	[100, 101, 102, 103]	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00	[10, 11, 12, 13]	[1, 2, 3, 4, 1, 2, 3, 4]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
-------------------------------------	-------------------------------	-----------------	---------------	---

	[100,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	[10,	
	101, 102,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	11, 12,	[1, 2, 3, 4, 5,
1050	103, 104]	22.00 23.00 24.00	13, 14]	1, 2, 3, 4, 5]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
	[100,		[10,	
	101, 102,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	11, 12,	[1, 2, 3, 4, 5,
	103, 104,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	13, 14,	6, 1, 2, 3, 4,
1060	105]	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00	15]	5, 6]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
	[100,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	[10,	
	101, 102,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	11, 12,	[1, 2, 3, 4, 5,
	103, 104,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	13, 14,	6, 7, 1, 2, 3,
1070	105, 106]	32.00 33.00 34.00	15, 16]	4, 5, 6, 7]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
	[100,		[10,	
	101, 102,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	11, 12,	[1, 2, 3, 4, 5,
	103, 104,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	13, 14,	6, 7, 8, 1, 2,
	105, 106,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	15, 16,	3, 4, 5, 6, 7,
1080	107]	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00	17]	8]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
	[100,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	[10,	[1, 2, 3, 4, 5,
1090	101, 102,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	11, 12,	6, 7, 8, 9, 1,

103, 104,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	13, 14,	2, 3, 4, 5, 6,
105, 106,	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00	15, 16,	7, 8, 9]
107, 108]	42.00 43.00 44.00	17, 18]	

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
1100	[100,		[10,	
	101, 102,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	11, 12,	
	103, 104,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	13, 14,	[1, 2, 3, 4, 5,
	105, 106,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	15, 16,	6, 7, 8, 9, 10,
	107, 108,	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00	17, 18,	1, 2, 3, 4, 5,
	109]	42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00	19]	6, 7, 8, 9, 10]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
1110	[100,		[10,	
	101, 102,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	11, 12,	[1, 2, 3, 4, 5,
	103, 104,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	13, 14,	6, 7, 8, 9, 10,
	105, 106,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	15, 16,	11, 1, 2, 3, 4,
	107, 108,	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00	17, 18,	5, 6, 7, 8, 9,
	109,	42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00	19, 20]	10, 11]
	110]	52.00 53.00 54.00		

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
1120	[100,		[10,	
	101, 102,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	11, 12,	[1, 2, 3, 4, 5,
	103, 104,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	13, 14,	6, 7, 8, 9, 10,
	105, 106,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	15, 16,	11, 12, 1, 2,
	107, 108,	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00	17, 18,	3, 4, 5, 6, 7,
	109, 110,	42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00	19, 20,	8, 9, 10, 11,
	111]	52.00 53.00 54.00 55.00 56.00 57.00 58.00 59.00	21]	12]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations

	[100,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	[10,
	101, 102,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	11, 12, [1, 2, 3, 4, 5,
	103, 104,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	13, 14, 6, 7, 8, 9, 10,
	105, 106,	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00	15, 16, 11, 12, 13, 1,
	107, 108,	42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00	17, 18, 2, 3, 4, 5, 6,
	109, 110,	52.00 53.00 54.00 55.00 56.00 57.00 58.00 59.00 60.00 61.00	19, 20, 7, 8, 9, 10,
1130	111, 112]	62.00 63.00 64.00	21, 22] 11, 12, 13]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
-------------------------------------	-------------------------------	-----------------	---------------	---

	[100, 101,	0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00	[10,
	102, 103,	12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00	11, 12, [1, 2, 3, 4, 5,
	104, 105,	22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00	13, 14, 6, 7, 8, 9, 10,
	106, 107,	32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00	15, 16, 11, 12, 13,
	108, 109,	42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00	17, 18, 14, 1, 2, 3, 4,
	110, 111,	52.00 53.00 54.00 55.00 56.00 57.00 58.00 59.00 60.00 61.00	19, 20, 5, 6, 7, 8, 9,
	112, 113]	62.00 63.00 64.00 65.00 66.00 67.00 68.00 69.00	21, 22, 10, 11, 12,
1140			23] 13, 14]

Total Within-Cluster Sum of Squares	Within-Cluster Sum of Squares	Cluster Centers	Cluster Sizes	Cluster Assignments for First 20 Observations
-------------------------------------	-------------------------------	-----------------	---------------	---

		0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00	
	[100, 101,	11.00 12.00 13.00 14.00 15.00 16.00 17.00 18.00 19.00	[10, 11, [1, 2, 3, 4, 5,
	102, 103,	20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00	12, 13, 6, 7, 8, 9, 10,
	104, 105,	29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00	14, 15, 11, 12, 13,
	106, 107,	38.00 39.00 40.00 41.00 42.00 43.00 44.00 45.00 46.00	16, 17, 14, 15, 1, 2,
	108, 109,	47.00 48.00 49.00 50.00 51.00 52.00 53.00 54.00 55.00	18, 19, 3, 4, 5, 6, 7,
	110, 111,	56.00 57.00 58.00 59.00 60.00 61.00 62.00 63.00 64.00	20, 21, 8, 9, 10, 11,
	112, 113,	65.00 66.00 67.00 68.00 69.00 70.00 71.00 72.00 73.00	22, 23, 12, 13, 14,
1150	114]	74.00	24] 15]

b) Wizualizacja wyników dla k = 2 do 7

```
par(mfrow = c(3, 2))
```

```
for (k in 2:7) {
```

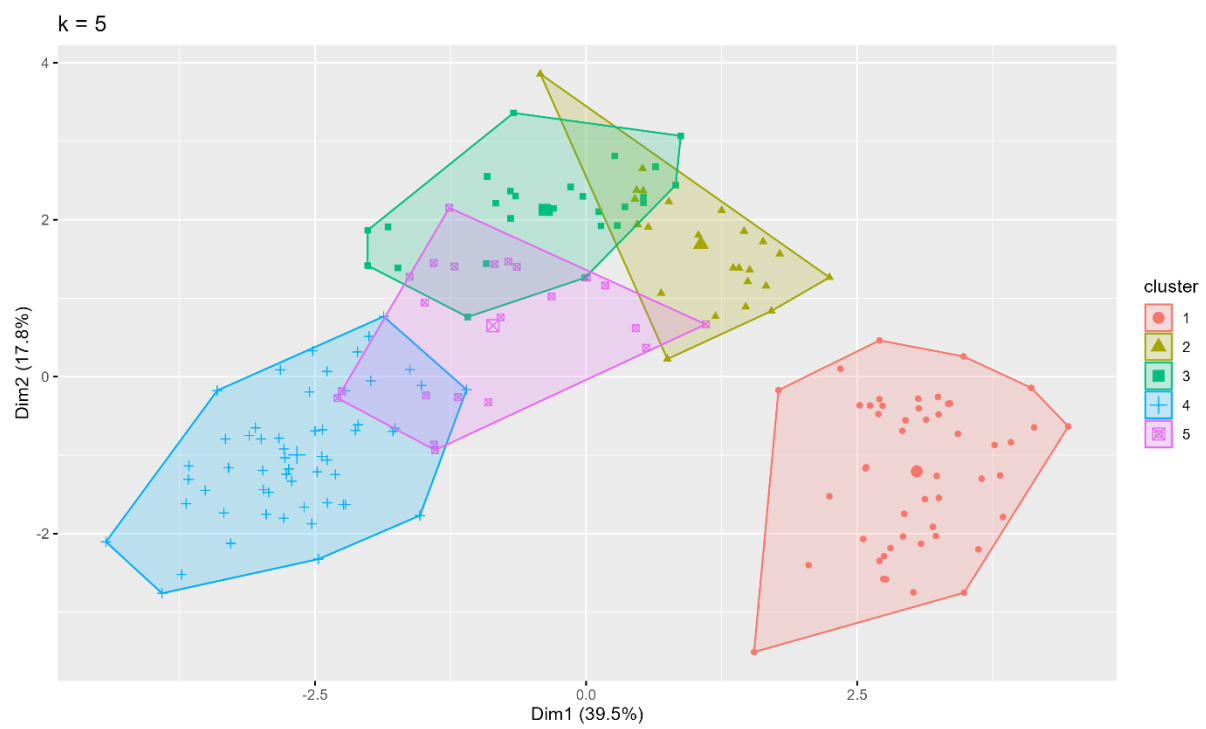
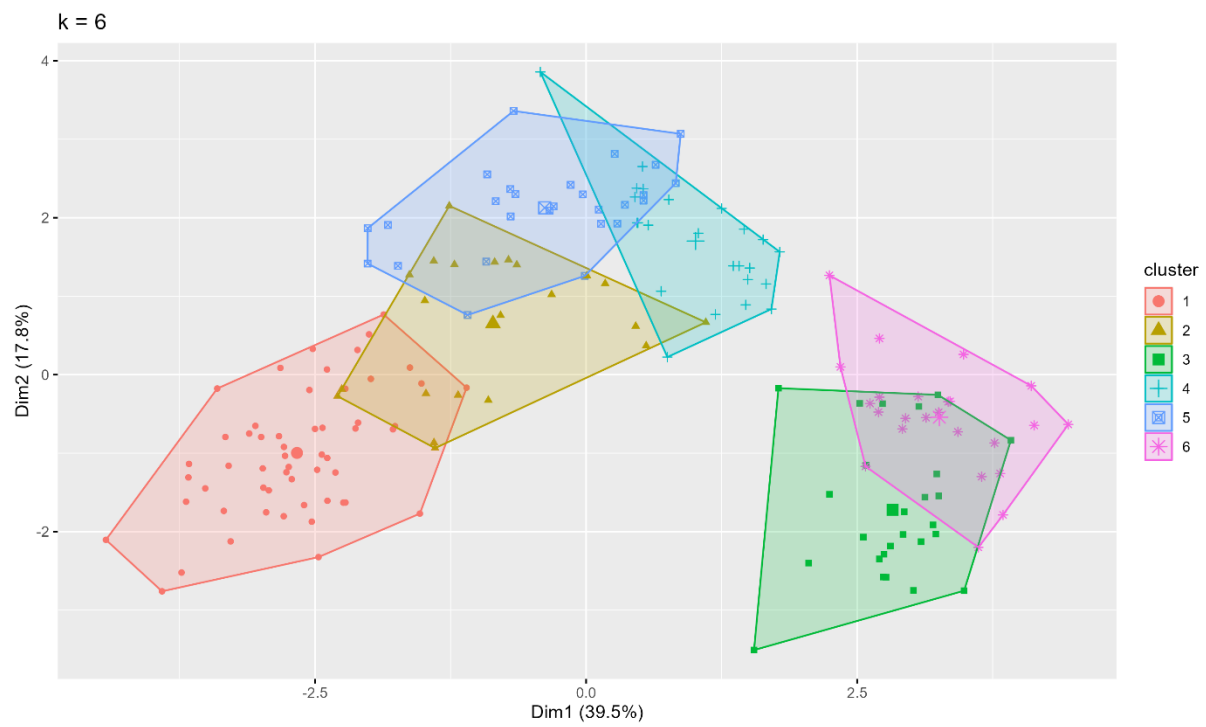
```
  # Tworzenie wykresu
```

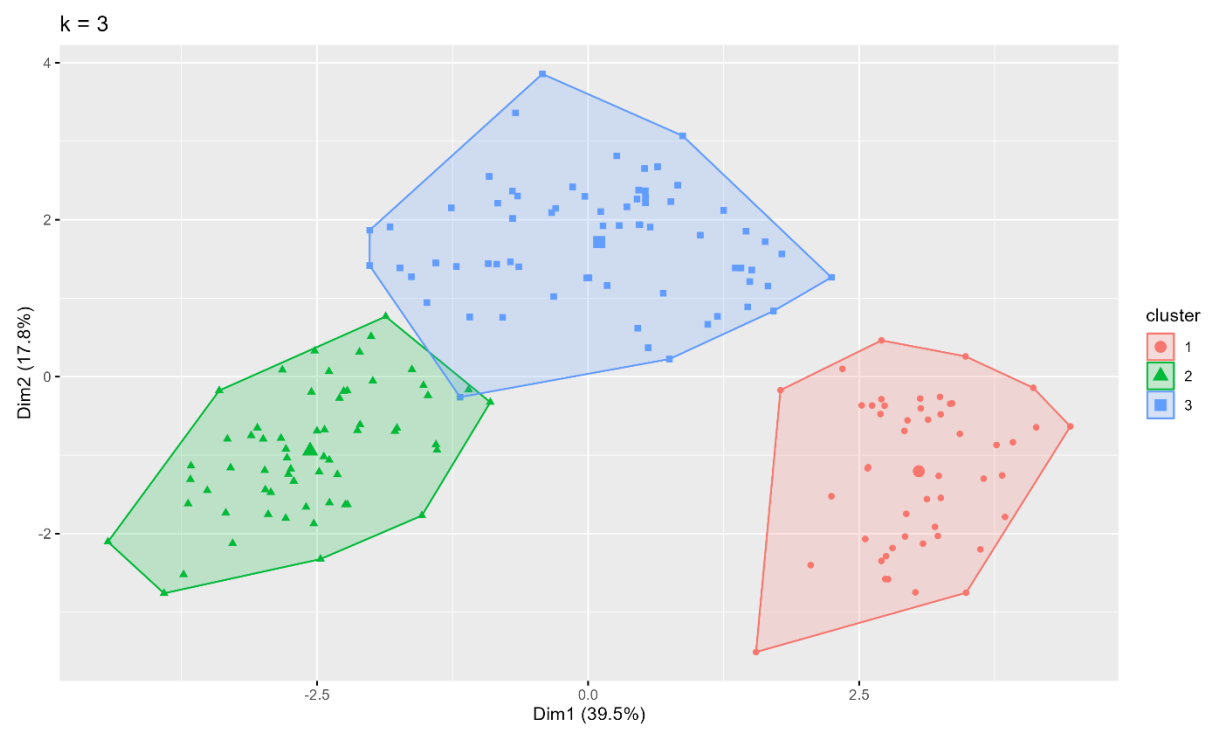
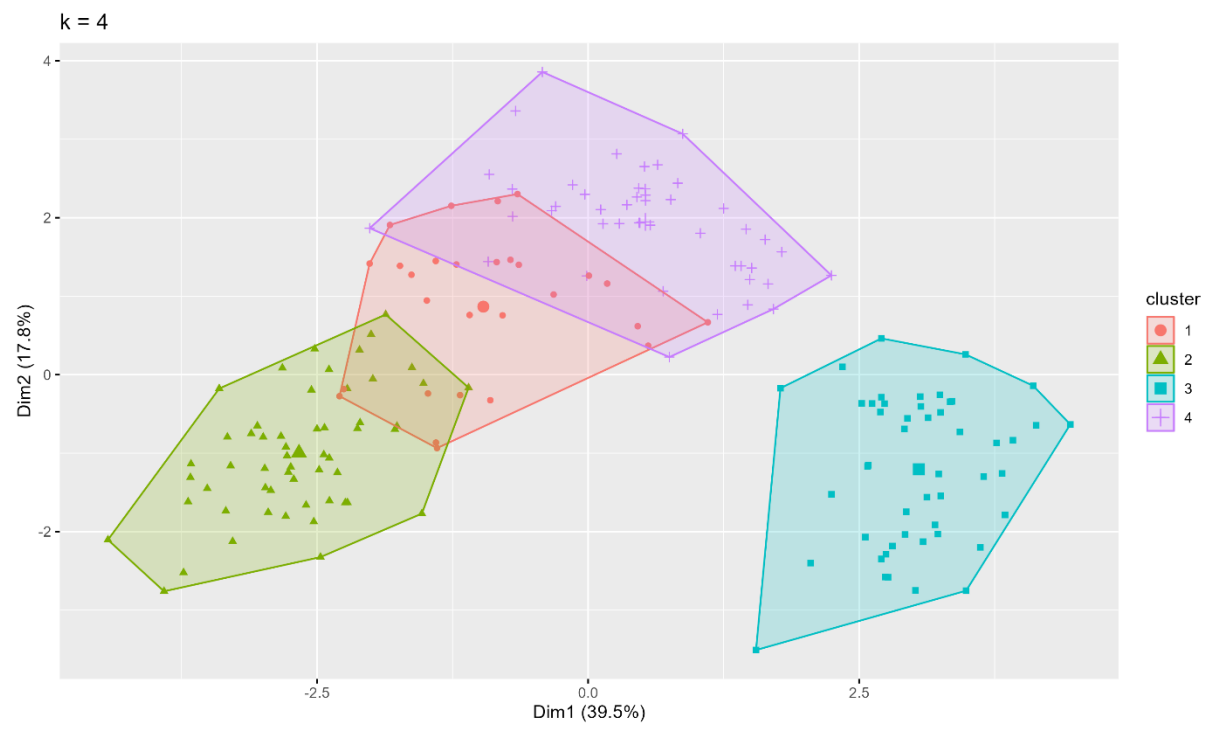
```
  plot <- fviz_cluster(results[[k]], data = wine_data_scaled, geom = "point", ellipse = TRUE, main = paste("k =", k))
```

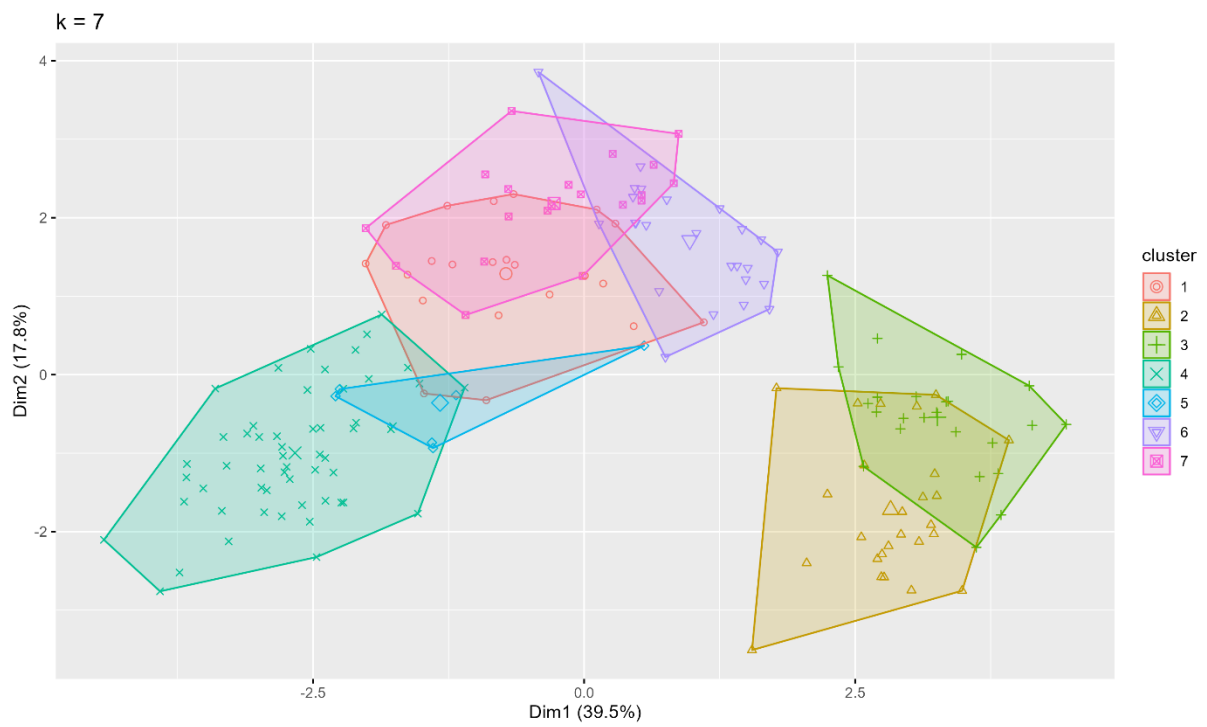
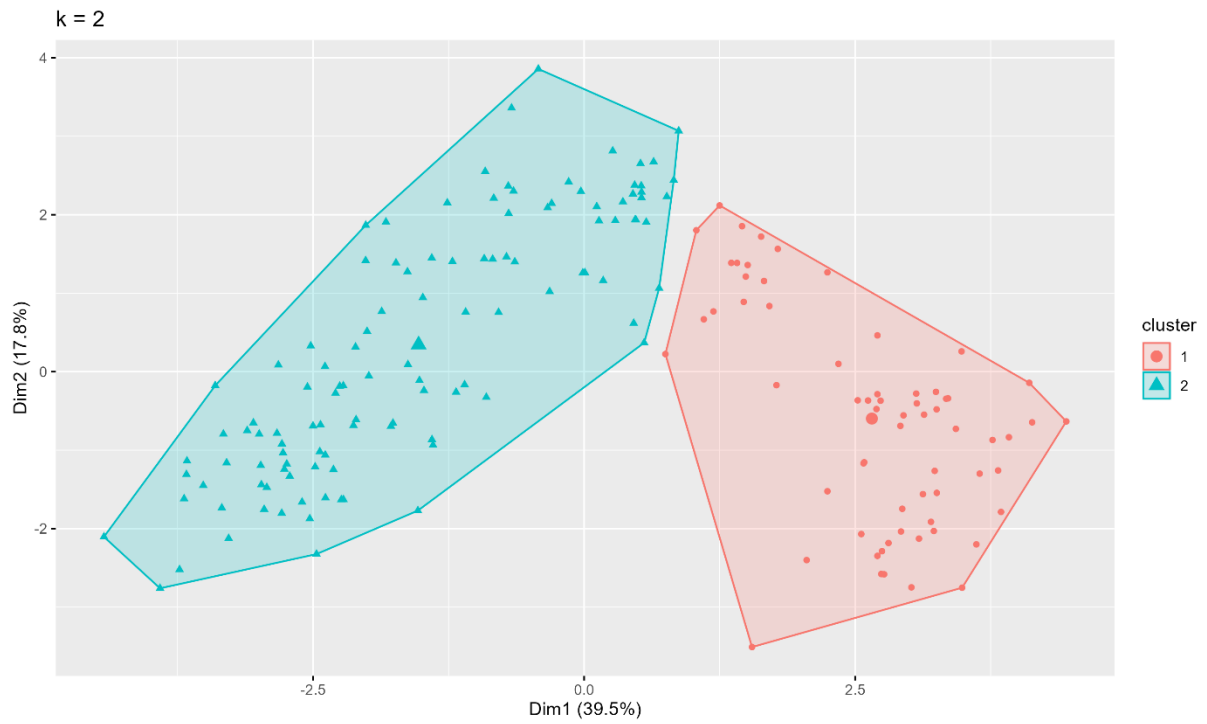
```
# Wyświetlanie wykresu
print(plot)

# Zapisywanie wykresu do pliku PNG
ggsave(filename = paste("kmeans_k", k, ".png"), plot = plot, width
= 10, height = 6)

# Sprawdź, czy plik został zapisany
if (file.exists(paste("kmeans_k", k, ".png"))) {
  cat("Plik", paste("kmeans_k", k, ".png"), "został zapisany.\n")
} else {
  cat("Plik", paste("kmeans_k", k, ".png"), "NIE został
zapisany.\n")
}
}
```



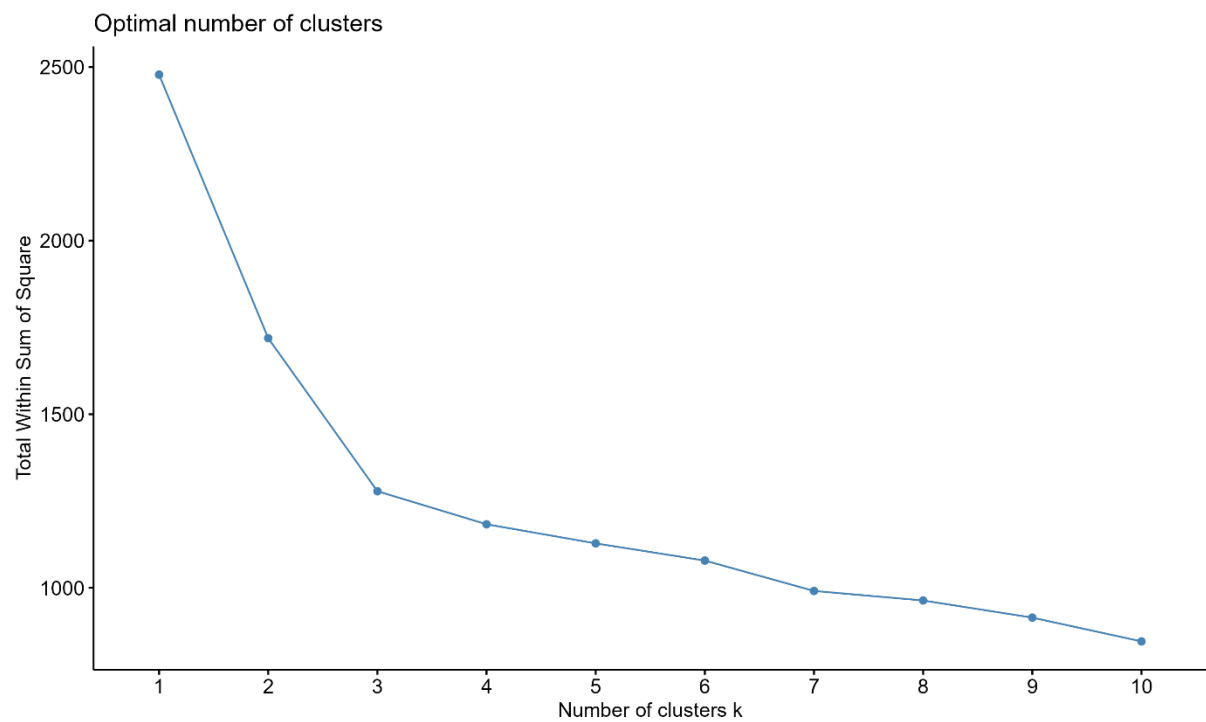




c) Określenie optymalnej liczby klastrów

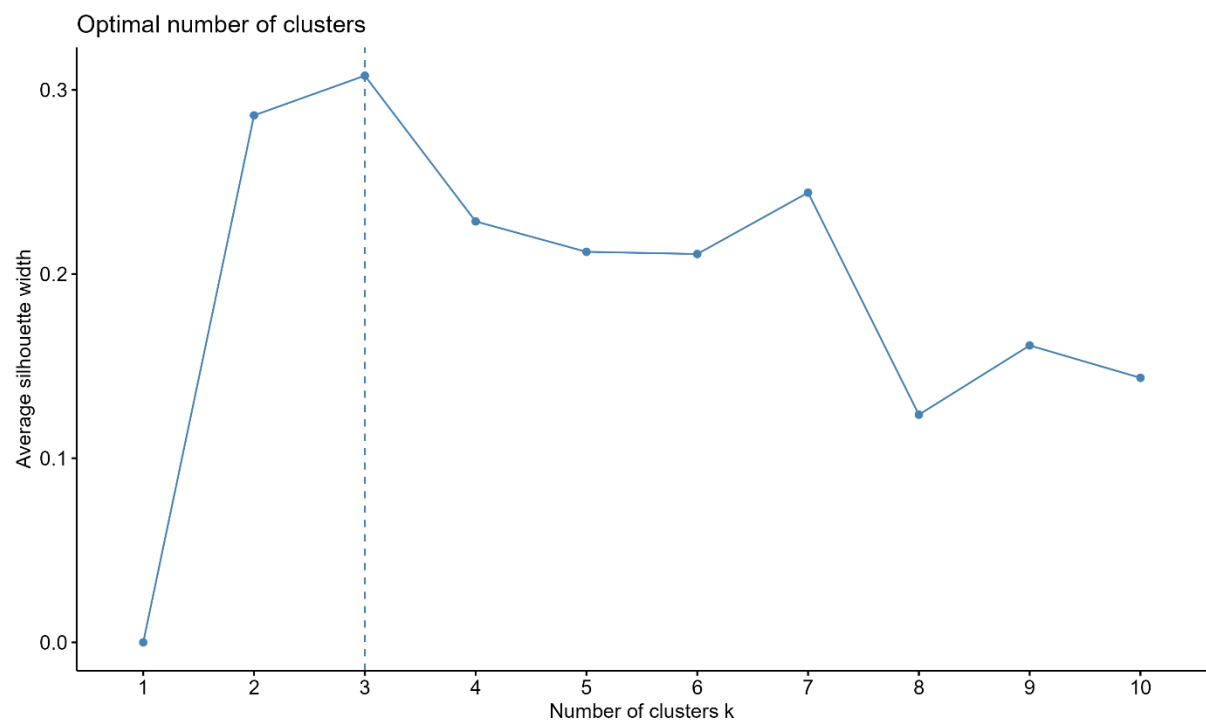
Metoda łokcia

```
fviz_nbclust(wine_data_scaled, kmeans, method = "wss")
```



Metoda średniego konturu

fviz_nbclust(wine_data_scaled, kmeans, method = "silhouette")

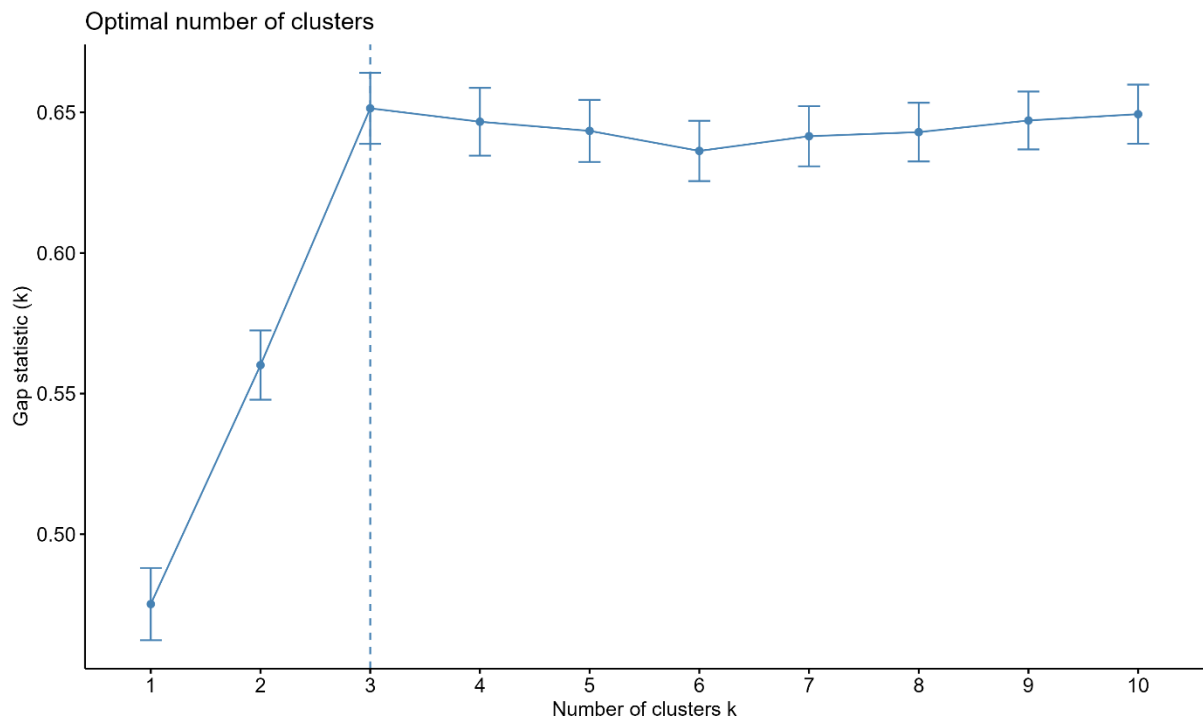


Statystyka luk

```
set.seed(123)
```

```
gap_stat <- clusGap(wine_data_scaled, FUN = kmeans, nstart = 25, K.max = 10, B = 50)
```

```
fviz_gap_stat(gap_stat)
```



Wnioski

1. **Metoda łokcia (Elbow Method):** Wykres pokazuje całkowitą sumę kwadratów wewnątrz klastrow dla różnych liczby klastrow k. Punkt, w którym krzywa zaczyna się wyginać (łokieć) i staje się mniej stroma, wskazuje na optymalną liczbę klastrow. Na załączonym wykresie punkt łokcia wydaje się znajdować przy k=3, co sugeruje, że 3 klastry mogą być odpowiednią liczbą dla tego zbioru danych.
2. **Metoda średniego konturu (Silhouette Method):** Wykres przedstawia średnią szerokość sylwetki dla różnych liczby klastrow. Wartość ta mierzy, jak dobrze pasuje próbek do swojego klastra (spójność) w porównaniu do innych klastrow (oddzielenie). Wyższa wartość średniej szerokości sylwetki wskazuje na lepsze dopasowanie. Na załączonym wykresie maksymalna średnia szerokość sylwetki pojawia się dla k=2.
3. **Statystyka luk (Gap Statistic):** Ta metoda porównuje logarytmiczny wskaźnik wewnątrz-sumy kwadratów dla różnych liczby klastrow z ich oczekiwanymi wartościami pod względem danych referencyjnych. Optymalną liczbę klastrow wskazuje największa luka. Na załączonym wykresie widoczny jest wyraźny szczyt dla k=3, co sugeruje, że najlepszą liczbą klastrow może być właśnie 3.

Dyskusja i wybór optymalnej liczby klastrow:

- Dwie z trzech metod (metoda łokcia i statystyka luk) sugerują, że optymalna liczba klastrów wynosi 3.
- Metoda średniego konturu wskazuje na 2 klastry, jednak wartość średniej szerokości sylwetki nie jest znacznie wyższa dla $k=2$ w porównaniu do $k=3$.

Biorąc pod uwagę powyższe informacje, i zakładając, że liczba klas (ground-truth labels) w zbiorze danych wine wynosi 3 (co jest typowe dla tego zbioru danych, zawierającego klasyfikację win do trzech różnych kultivarów), wybór 3 klastrów wydaje się być najbardziej odpowiedni. Jest to zgodne z rzeczywistą liczbą klas w danych, co dodatkowo potwierdza wybór.

Lista 9

Przeprowadź dyskretyzację wybranej zmiennej numerycznej stosując dwie różne metody: według równej szerokości oraz według równej częstości. a) Dla każdego przypadku (metody) zamieść dwa wykresy: rozkład nowych wartości oraz histogram dla starych wartości wraz z liniami dzielącymi przedziały. b) Porównaj i przedyskutuj wyniki działania obu metod.

```
# Ustawienie ścieżki dostępu do danych

path <- "C:/Users/petit/Desktop/repos/UO/rok 3/Wprowadzenie do
eksploracji danych/lista8"

setwd(path)

# Read the data

wine <- read.csv('wine/wine.data', header = FALSE)

## 1. Dyskretyzacja zmiennej numerycznej

# Załóżmy, że chcesz dyskretyzować drugą kolumnę numeryczną (V2)
variable_to_discretize <- wine[, 2] # druga kolumna

# Ustal liczbę przedziałów, np. 5
number_of_bins <- 5

# Dyskretyzacja według równej szerokości
equal_width_bins <- cut(variable_to_discretize, breaks =
number_of_bins, include.lowest = TRUE)

# Dyskretyzacja według równej częstości
library(classInt)

freq_breaks <- classIntervals(variable_to_discretize, n =
number_of_bins, style = "quantile")$brks

equal_freq_bins <- cut(variable_to_discretize, breaks = freq_breaks,
include.lowest = TRUE)
```

```
# Wykresy dla dyskretyzacji według równej szerokości

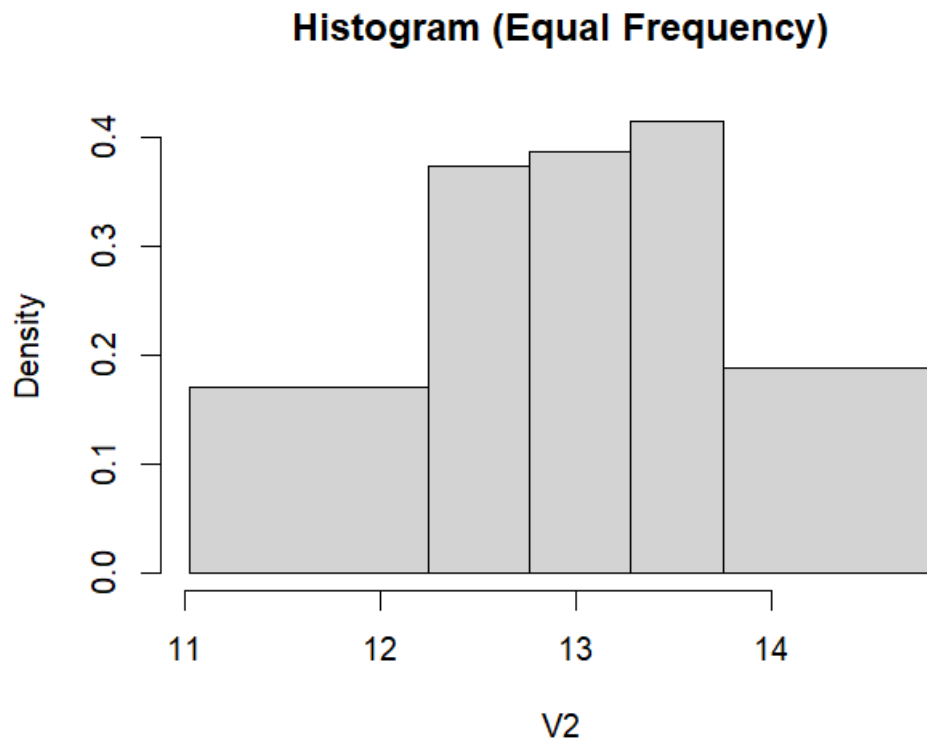
hist(variable_to_discretize, breaks = number_of_bins, main =
"Histogram (Equal Width)", xlab = "V2")

rug(jitter(as.numeric(equal_width_bins)))

# Wykresy dla dyskretyzacji według równej częstości

hist(variable_to_discretize, breaks = freq_breaks, main = "Histogram
(Equal Frequency)", xlab = "V2")

rug(jitter(as.numeric(equal_freq_bins)))
```



Wnioski

1. Słupki histogramu przedstawiają gęstość, a nie liczbę obserwacji. Gęstość jest używana, gdy zmienna została podzielona na przedziały o różnej szerokości.
2. Widoczne są cztery przedziały, co sugeruje, że algorytm wydzielił dane do czterech równolicznych grup. To jest zgodne z liczbą przedziałów określoną w parametrze **number_of_bins**.

3. Środkowe słupki są szersze niż skrajne, co wskazuje, że wartości w środkowych przedziałach są bardziej skupione, a zakresy tych przedziałów są szersze, aby pomieścić podobną liczbę obserwacji, co w innych przedziałach.
4. Szerokość słupków histogramu nie jest jednakowa, co jest charakterystyczne dla dyskretyzacji według równej częstości. Każdy słupek reprezentuje przedział o zbliżonej liczbie obserwacji, ale przedziały te mogą mieć różną szerokość.
5. Nierówna szerokość słupków może wskazywać na to, że rozkład oryginalnej zmiennej nie jest jednorodny. Na przykład, możemy zaobserwować, że dane są bardziej skoncentrowane w określonych zakresach wartości.
6. Ostatni słupek po prawej stronie jest znacznie niższy niż pozostałe, co oznacza, że przedział ten jest szerszy niż pozostałe przedziały, aby zawierał taką samą liczbę obserwacji jak one.

Przeprowadź dyskretyzację wszystkich numerycznych zmiennych w tabeli, stosując wybraną metodę (spośród: "interval", "frequency", "cluster"). a) Wyświetl wartości przed i po dyskretyzacji dla kilku pierwszych obserwacji/wierszy.

```
> ## 2. Dyskretyzacja wszystkich numerycznych zmiennych
```

```
>
```

```
> # Dyskretyzacja wszystkich numerycznych zmiennych metodą "frequency"
```

```
> discretized_wine <- wine
```

```
> for(i in 2:ncol(wine)) {
```

```
+   discretized_wine[, i] <- cut(wine[, i], breaks = classIntervals(wine[, i], n = number_of_bins, style = "quantile")$brks, include.lowest = TRUE)
```

```
+ }
```

```
>
```

```
> # Wyświetlenie wartości przed i po dyskretyzacji dla kilku pierwszych wierszy
```

```
> head(wine)
```

```
  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14
1  1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92 1065
2  1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40 1050
3  1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17 1185
4  1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45 1480
5  1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93 735
6  1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85 1450
```

```
> head(discretized_wine)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	1	(13.8,14.8]	(1.51,1.73]	(2.42,2.61]	[10.6,16.8]	(111,162]	(2.53,2.86]	(2.98,5.08]	(0.26,0.3]	(1.99,3.58]
2	1	(12.8,13.3]	(1.73,2.13]	[1.36,2.18]	[10.6,16.8]	(94.8,101]	(2.53,2.86]	(2.46,2.98]	[0.13,0.26]	(1.1,1.42]
3	1	(12.8,13.3]	(2.13,3.41]	(2.61,3.23]	(18.6,20]	(94.8,101]	(2.53,2.86]	(2.98,5.08]	(0.26,0.3]	(1.99,3.58]
4	1	(13.8,14.8]	(1.73,2.13]	(2.42,2.61]	[10.6,16.8]	(111,162]	(2.86,3.88]	(2.98,5.08]	[0.13,0.26]	(1.99,3.58]
5	1	(12.8,13.3]	(2.13,3.41]	(2.61,3.23]	(20,22]	(111,162]	(2.53,2.86]	(2.46,2.98]	(0.3,0.39]	(1.66,1.99]
6	1	(13.8,14.8]	(1.73,2.13]	(2.42,2.61]	[10.6,16.8]	(111,162]	(2.86,3.88]	(2.98,5.08]	(0.3,0.39]	(1.66,1.99]

	V11	V12	V13	V14
1	(5.28,6.99]	(0.91,1.04]	(3.26,4]	(1.05e+03,1.68e+03]
2	(4.08,5.28]	(1.04,1.16]	(3.26,4]	(1.05e+03,1.68e+03]
3	(5.28,6.99]	(0.91,1.04]	(2.9,3.26]	(1.05e+03,1.68e+03]
4	(6.99,13]	(0.74,0.91]	(3.26,4]	(1.05e+03,1.68e+03]
5	(4.08,5.28]	(0.91,1.04]	(2.9,3.26]	(606,742]
6	(5.28,6.99]	(1.04,1.16]	(2.52,2.9]	(1.05e+03,1.68e+03]

```
> # Wyświetlenie wartości przed i po dyskretyzacji dla kilku pierwszych wierszy
> head(wine)
  V1  V2  V3  V4  V5  V6  V7  V8  V9  V10  V11  V12  V13  V14
1  1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92 1065
2  1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40 1050
3  1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17 1185
4  1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45 1480
5  1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93 735
6  1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85 1450
> head(discretized_wine)
  V1  V2  V3  V4  V5  V6  V7  V8  V9  V10
1  1 (13.8,14.8] (1.51,1.73] (2.42,2.61] [10.6,16.8] (111,162] (2.53,2.86] (2.98,5.08] (0.26,0.3] (1.99,3.58]
2  1 (12.8,13.3] (1.73,2.13] [1.36,2.18] [10.6,16.8] (94.8,101] (2.53,2.86] (2.46,2.98] [0.13,0.26] (1.1,1.42]
3  1 (12.8,13.3] (2.13,3.41] (2.61,3.23] (18.6,20] (94.8,101] (2.53,2.86] (2.98,5.08] (0.26,0.3] (1.99,3.58]
4  1 (13.8,14.8] (1.73,2.13] (2.42,2.61] [10.6,16.8] (111,162] (2.86,3.88] (2.98,5.08] [0.13,0.26] (1.99,3.58]
5  1 (12.8,13.3] (2.13,3.41] (2.61,3.23] (20,22] (111,162] (2.53,2.86] (2.46,2.98] (0.3,0.39] (1.66,1.99]
6  1 (13.8,14.8] (1.73,2.13] (2.42,2.61] [10.6,16.8] (111,162] (2.86,3.88] (2.98,5.08] (0.3,0.39] (1.66,1.99]
  V11  V12  V13  V14
1 (5.28,6.99] (0.91,1.04] (3.26,4] (1.05e+03,1.68e+03]
2 (4.08,5.28] (1.04,1.16] (3.26,4] (1.05e+03,1.68e+03]
3 (5.28,6.99] (0.91,1.04] (2.9,3.26] (1.05e+03,1.68e+03]
4 (6.99,13] (0.74,0.91] (3.26,4] (1.05e+03,1.68e+03]
5 (4.08,5.28] (0.91,1.04] (2.9,3.26] (606,742]
6 (5.28,6.99] (1.04,1.16] (2.52,2.9] (1.05e+03,1.68e+03]
```


Omówienie wyników

1. **Zakresy przedziałów:** Oryginalne wartości dla V2 pokazują dokładne pomiary zawartości alkoholu, podczas gdy po dyskretyzacji zostały one zgrupowane w kategorie. Na przykład, wartość 14.23 została zaklasyfikowana do przedziału (13.8, 14.8], co oznacza, że wszystkie wina z zawartością alkoholu w tym przedziale będą traktowane jako identyczne pod tym względem w dalszej analizie.
2. **Strata informacji:** Proces dyskretyzacji nieuchronnie prowadzi do pewnego stopnia utraty informacji. Zamiast precyzyjnej wartości, mamy teraz kategorię, która jest mniej szczegółowa. To może być korzystne dla pewnych rodzajów analiz, takich jak analiza reguł asocjacyjnych, ale mniej korzystne dla innych, które wymagają większej precyzji.
3. **Uproszczenie analizy:** Kategoryzacja danych może uprościć niektóre analizy statystyczne i ułatwić wizualizację oraz interpretację danych. Na przykład, moglibyśmy łatwo porównać liczby win z różnych przedziałów zawartości alkoholu.
4. **Wpływ na modele:** Dyskretyzacja może wpływać na wydajność modeli predykcyjnych. Dla niektórych algorytmów uczenia maszynowego, takich jak drzewa decyzyjne, dyskretyzacja może być korzystna, podczas gdy dla innych, takich jak regresja liniowa, może to negatywnie wpłynąć na wydajność modelu.

3. Do danych w tabeli po dyskretyzacji zastosuj algorytm A priori, podając zadane wartości minimalnego wsparcia i ufności. a) Napisz, ile reguł zostało wygenerowanych? b) Wyświetl kilka pierwszych reguł, posortowanych według miary lift. c) Zilustruj je na wykresie. d) Zinterpretuj "w języku naturalnym" pierwszą regułę. e) Przedyskutuj uzyskane wyniki.

3. Algorytm A priori

```
# Parametry algorytmu A priori
```

```
min_support <- 0.1
```

```
min_confidence <- 0.8
```

```
library(arules)
```

```
data <- as(discretized_wine, "transactions")
```

```
rules <- apriori(data, parameter = list(support = min_support, confidence = min_confidence))
```

```
# Ilość wygenerowanych reguł
```

```
length(rules)
```

```
# Sortowanie reguł według miary lift
```

```
rules_sorted <- sort(rules, by = "lift", decreasing = TRUE)
```

```
# Wyświetlenie kilku pierwszych reguł
```

```
head(rules_sorted)
```

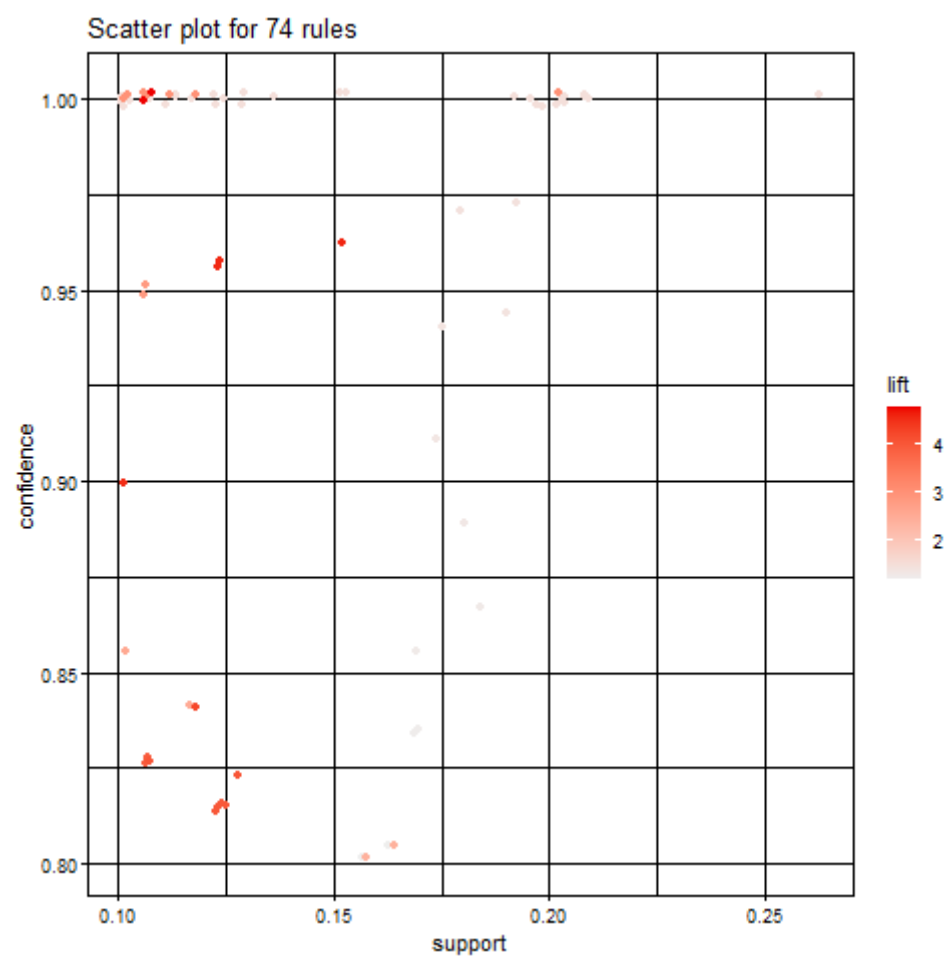
```
plot(rules, method = "scatter", measure = c("support", "confidence"))
```

```
plot(rules, method = "scatter", measure = c("lift", "support"))
```

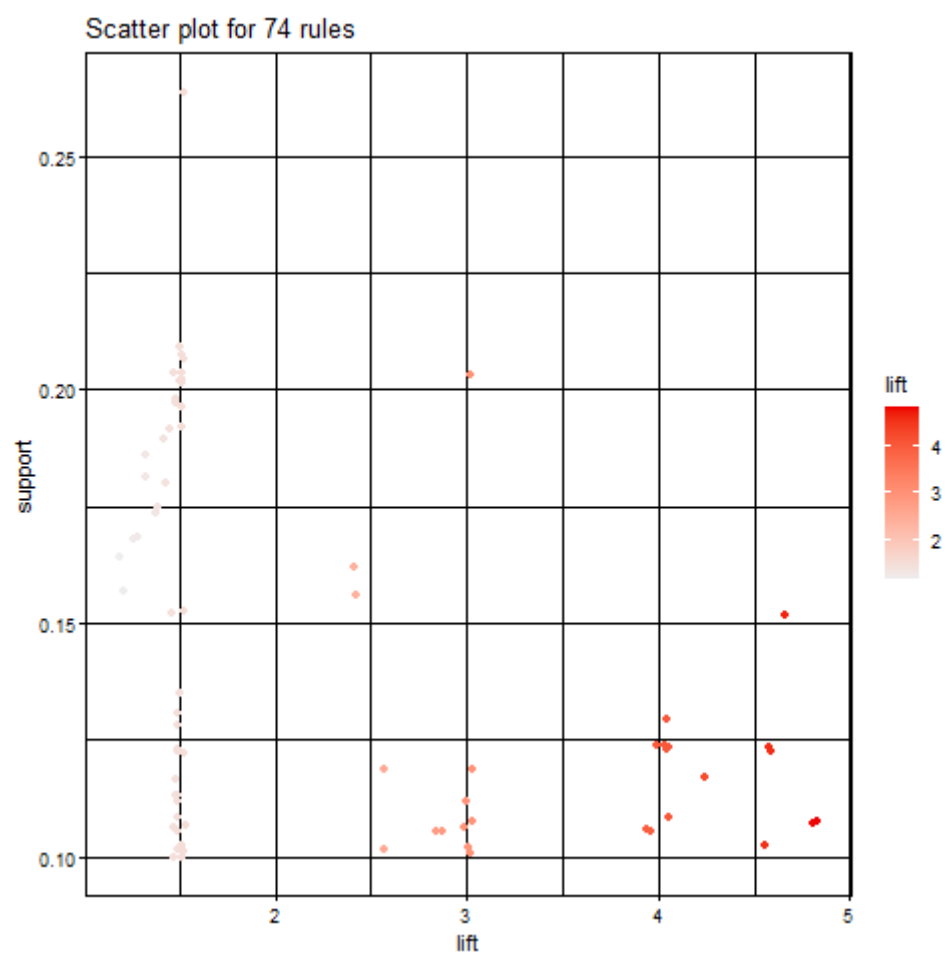
```
plot(rules, method = "scatter", measure = c("lift", "confidence"))
```

```
Only unique breaks are used reducing the number of intervals. Look at ?discretize for details.  
> rules <- apriori(data, parameter = list(support = min_support, confidence = min_confidence))  
Apriori  
  
Parameter specification:  
confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext  
0.8 0.1 1 none FALSE TRUE 5 0.1 1 10 rules TRUE  
  
Algorithmic control:  
filter tree heap memopt load sort verbose  
0.1 TRUE TRUE FALSE TRUE 2 TRUE  
  
Absolute minimum support count: 17  
  
set item appearances ...[0 item(s)] done [0.00s].  
set transactions ...[67 item(s), 178 transaction(s)] done [0.00s].  
sorting and recoding items ... [67 item(s)] done [0.00s].  
creating transaction tree ... done [0.00s].  
checking subsets of size 1 2 3 4 done [0.00s].  
writing ... [74 rule(s)] done [0.00s].  
creating S4 object ... done [0.00s].  
>
```

```
writing ... [74 rule(s)] done [0.00s].  
creating S4 object ... done [0.00s].  
>  
> # Ilość wygenerowanych reguł  
> length(rules)  
[1] 74  
>  
> # Sortowanie reguł według miary lift  
> rules_sorted <- sort(rules, by = "lift", decreasing = TRUE)  
>  
> # Wyświetlenie kilku pierwszych reguł  
> head(rules_sorted)  
set of 6 rules  
> plot(rules, method = "scatter", measure = c("support", "confidence"))  
To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.  
> plot(rules, method = "scatter", measure = c("lift", "support"))
```



Plot 1 wykres_wsparcie_vs_ufnosc



Plot 2 wykres_lift_vs_wsparcie

