Błażej Domagała - WED

# Lista 1

## Zadanie 1

a) Nazwa zbioru: Wine

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

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.

Obraz zawierający zrzut ekranu, tekst, Czcionka

Opis wygenerowany automatycznie

c) Liczba obserwacji w zbiorze: 178

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

e) Zmienna celu:

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

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

f) Wykaz i opis cech:

1. **Class**: Zmienna kategoryczna. 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.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

## Zadanie 2

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

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, zrzut ekranu

Opis wygenerowany automatycznie

1. Polecenie View (fragment print-screena z tabelką)

Obraz zawierający Czcionka, tekst, zrzut ekranu, linia

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, wyświetlacz, oprogramowanie

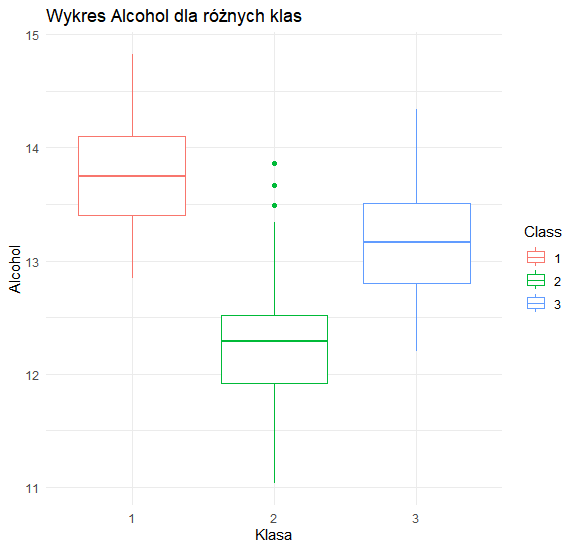
Opis wygenerowany automatycznie

1. Podsumowanie cech (summary)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

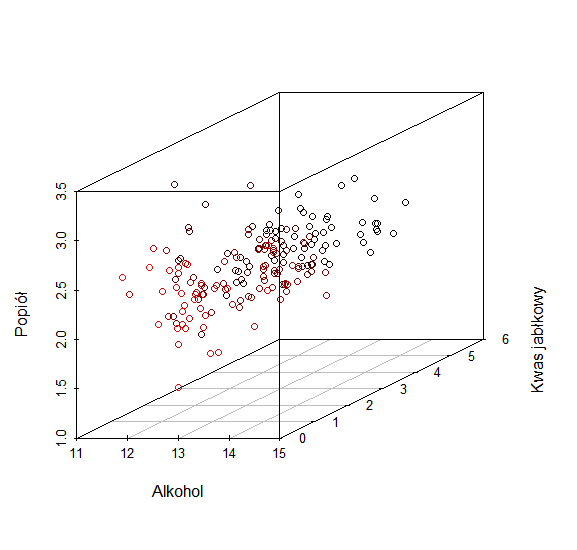
1. Wykres 2D ilustrujący wybraną cechę dla różnych klas



Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

1. Wykres 3D dla trzech wybranych cech (bez klasy)



Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

# Lista 2

1. 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]

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

1. 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,]

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

1. 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.")

}

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

1. 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")

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

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

# e) Macierz korelacji dla wybranych pięciu zmiennych i jej wizualizacja

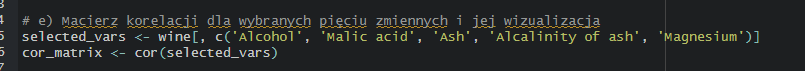
selected\_vars <-

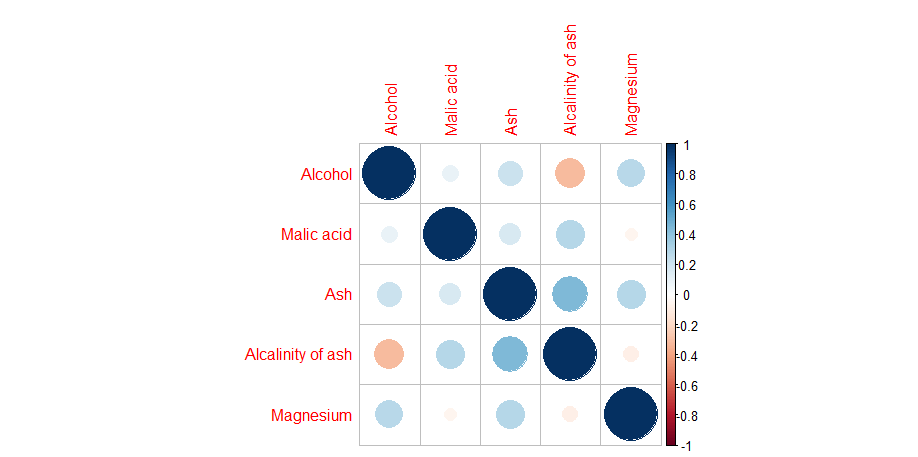
wine[, c('Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium')]

cor\_matrix <- cor(selected\_vars)

library(corrplot)

corrplot(cor\_matrix, method = "circle")





1. 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,

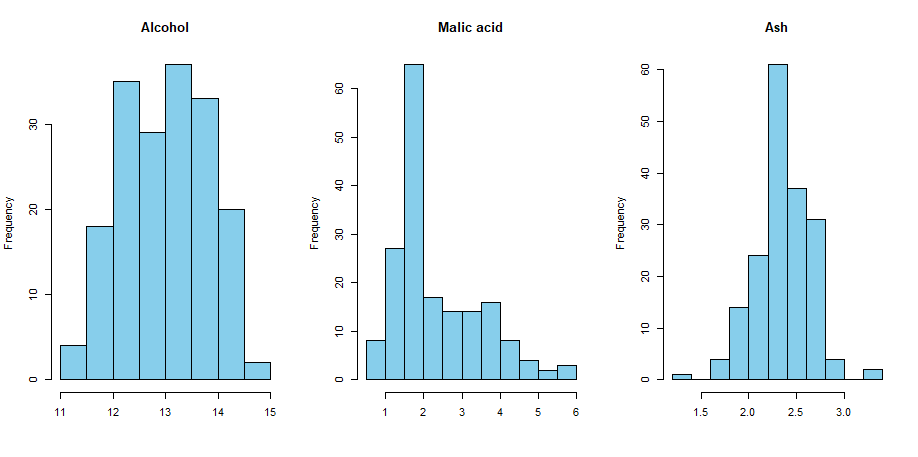
main = 'Ash',

xlab = '',

col = 'skyblue')

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie



# Lista 3

## 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'

)

Obraz zawierający zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne, tekst

Opis wygenerowany automatycznie

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

Obraz zawierający zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne, tekst

Opis wygenerowany automatycznie

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.

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

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

}

}

g) Zapisz przetworzone dane do pliku (razem z nowymi nazwami kolumn). Załącz fragment print-screena zawartości (początku) pliku.

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, komputer

Opis wygenerowany automatycznie

# Lista 4

## 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)

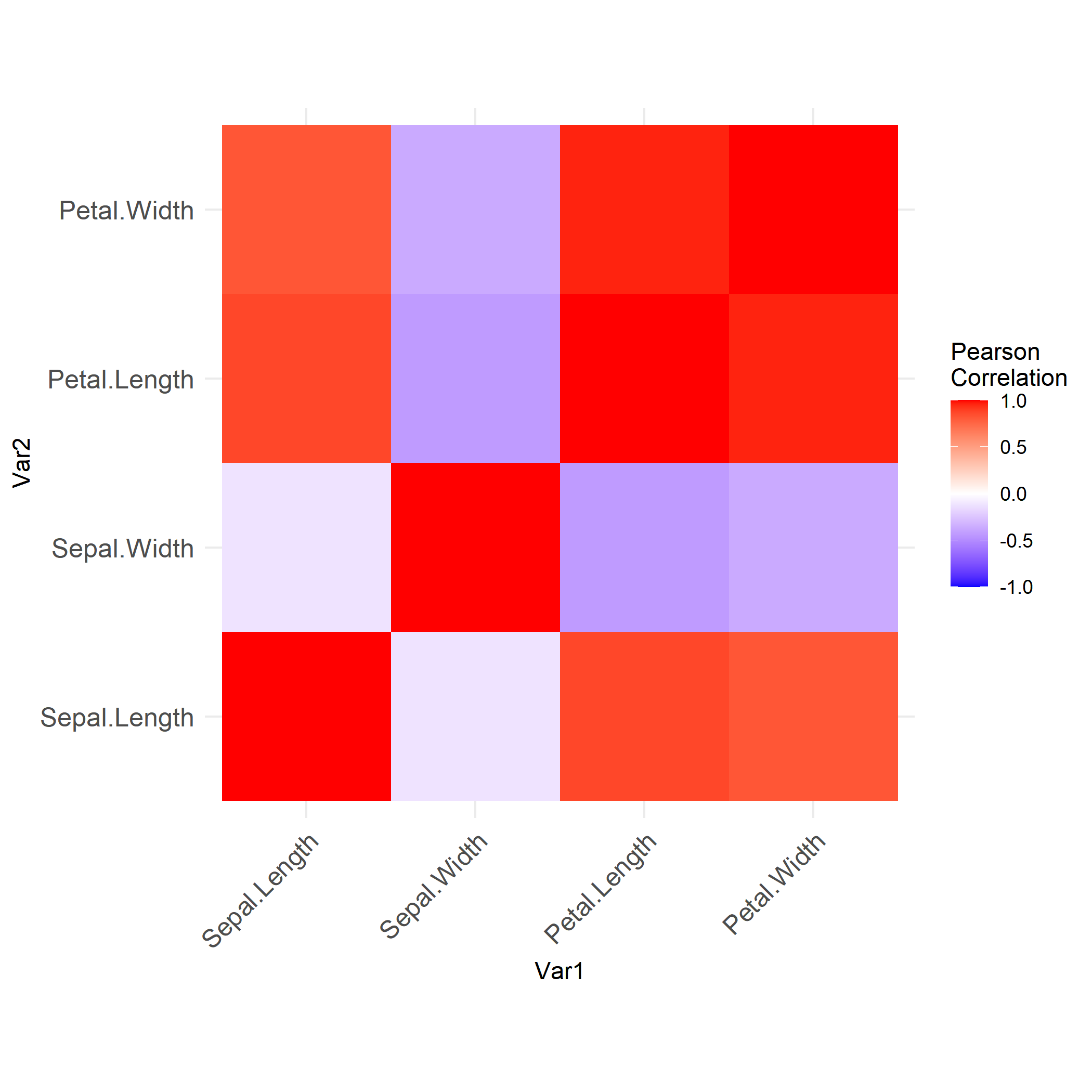
# Usunięcie kolumny Species (bo to jest nasza zmienna celu)

df <- iris[,-5]

# Użycie funkcji na df z progiem 0.5

correlation\_matrix(df, 0.5)

## 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:

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

# 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") +

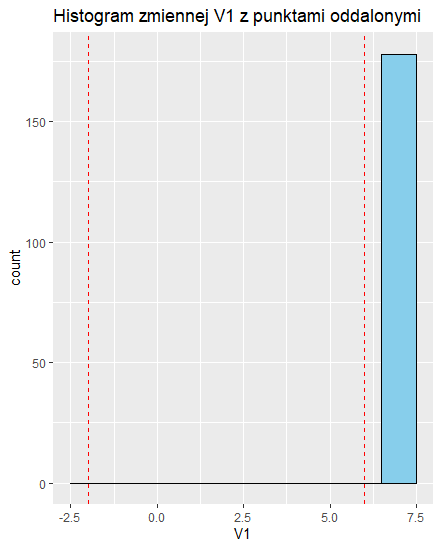
geom\_vline(xintercept = outliers\_info[[var]]$dolna\_granica, color="red", linetype="dashed") +

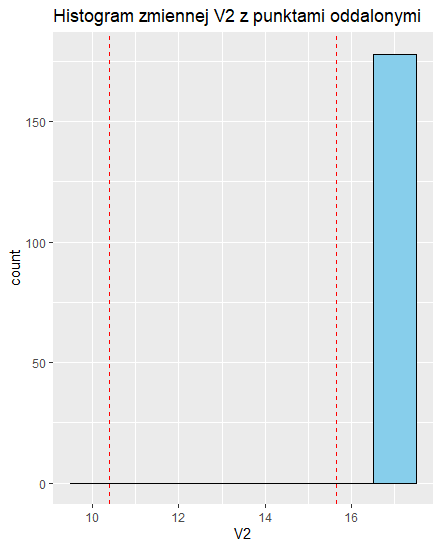
geom\_vline(xintercept = outliers\_info[[var]]$gorna\_granica, color="red", linetype="dashed") +

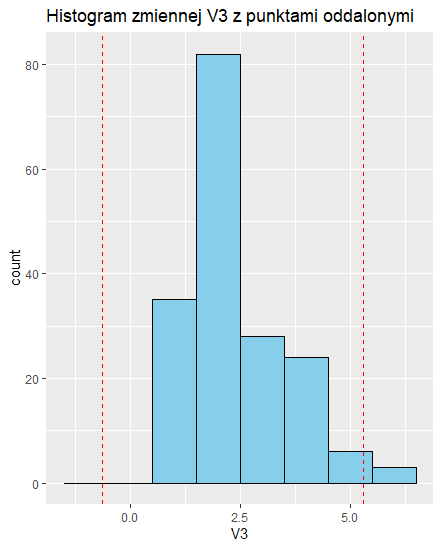
ggtitle(paste("Histogram zmiennej", var, "z punktami oddalonymi"))

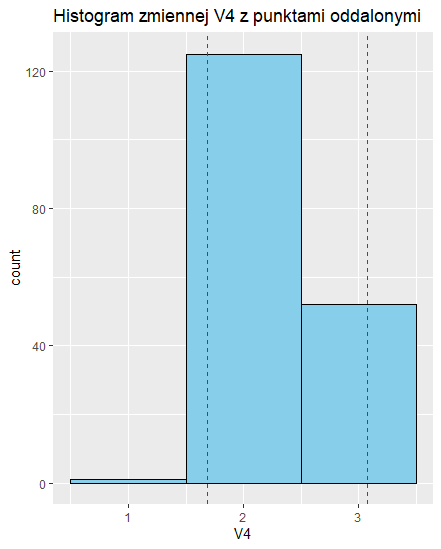
plots[[var]] <- plot

}









# 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 samych 4 zmiennych

cleaned\_plots <- list()

for (var in selected\_vars) {

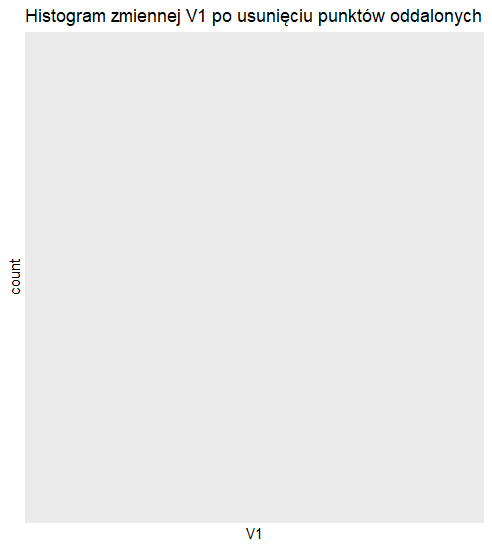
plot <- ggplot(cleaned\_wine\_data, aes\_string(x=var)) +

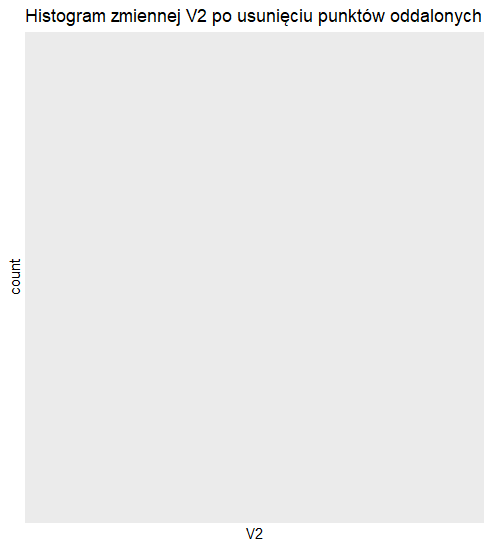
geom\_histogram(binwidth = 1, fill="green", color="black") +

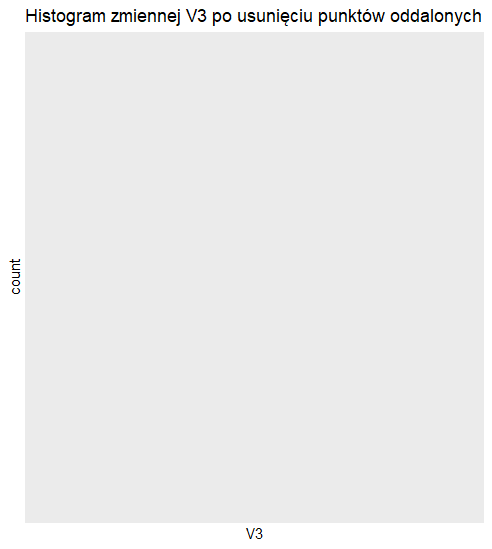
ggtitle(paste("Histogram zmiennej", var, "po usunięciu punktów oddalonych"))

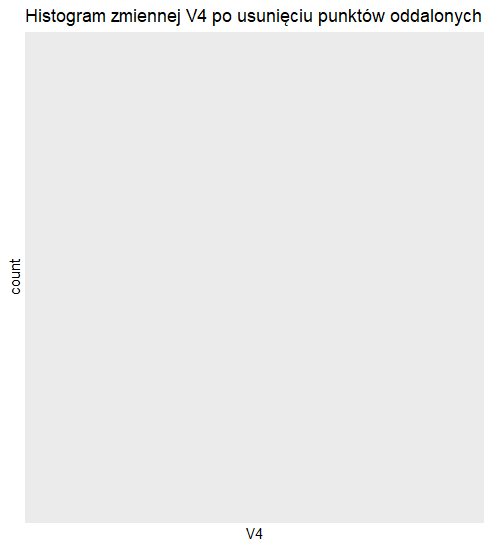
cleaned\_plots[[var]] <- plot

}









# 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

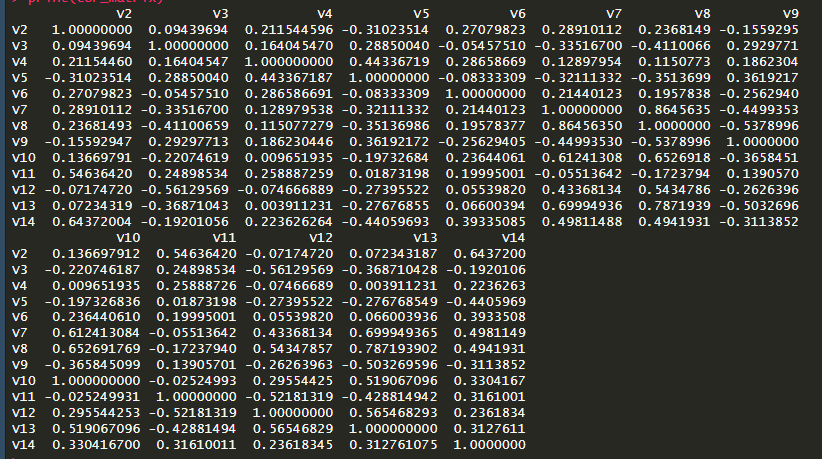
wine <- read.csv('wine\\wine.data', header = FALSE)

wine\_features <- wine[, -1]

# utworzenie macierzy korelacji

cor\_matrix <- cor(wine\_features)

print(cor\_matrix)



# 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)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# Utworzenie tablicy 2D z rangami:

ranks\_matrix <- data.frame(pearson = pearson\_rank)

rownames(ranks\_matrix) <- colnames(wine\_features)

print(ranks\_matrix)

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

# Dodanie kolumny ze średnią wartością rangi:

ranks\_matrix$average\_rank <- rowMeans(ranks\_matrix, na.rm = TRUE)

print(ranks\_matrix)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# Sortowanie cech według wartości średniej rangi:

sorted\_ranks <- ranks\_matrix[order(ranks\_matrix$average\_rank), ]

print(sorted\_ranks)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# 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)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# 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)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# 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)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# 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)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# 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)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# Chi-Square Test

# Uwaga: Ten test ma sens tylko dla cech kategorialnych.

# Aktualizacja średniej rangi

ranks\_matrix$average\_rank <- rowMeans(ranks\_matrix[, -1], na.rm = TRUE)

print(ranks\_matrix$average\_rank)

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

# Sortowanie cech według wartości średniej rangi

sorted\_ranks <- ranks\_matrix[order(ranks\_matrix$average\_rank), ]

print(sorted\_ranks)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

## Wnioski

1. **Ważność Cech wg Korelacji Pearsona:**
   * Cecha **V11** wykazuje najwyższą korelację (najniższą rangę) z etykietami klas, co wskazuje na jej potencjalnie dużą ważność 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 nagłówków.

# Wczytanie danych

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

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

Obraz zawierający tekst, zrzut ekranu, menu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

# b) Ładunki czynnikowe

loadings <- wine\_data\_pca$var$coord

print(loadings)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# c) Zasoby zmienności wspólnej

communalities <- wine\_data\_pca$var$cos2

print(communalities)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

# 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

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, diagram, linia, krąg

Opis wygenerowany automatycznie

# e) Wykres łącznego wkładu dla pierwszych trzech wymiarów

fviz\_contrib(wine\_data\_pca, choice = "var", axes = 1:3)

ggsave("contrib\_plot.png", bg = "white")

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

# f) Wektory własne

eigen\_vectors <- wine\_data\_pca$ind$coord

print(eigen\_vectors)

# g) Wybór liczby składowych

# Analiza procentu wyjaśnianej wariancji

fviz\_screeplot(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

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

# 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



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 2478 | [2478] | -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 | [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** |
| 1717,525 | [532.4956, 1185.03] | 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 | [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** |
| 1050 | [100, 101, 102, 103, 104] | 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 20.00 21.00 22.00 23.00 24.00 | [10, 11, 12, 13, 14] | [1, 2, 3, 4, 5, 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** |
| 1060 | [100, 101, 102, 103, 104, 105] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 | [10, 11, 12, 13, 14, 15] | [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 1070 | [100, 101, 102, 103, 104, 105, 106] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 | [10, 11, 12, 13, 14, 15, 16] | [1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 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** |
| 1080 | [100, 101, 102, 103, 104, 105, 106, 107] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 | [10, 11, 12, 13, 14, 15, 16, 17] | [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 1090 | [100, 101, 102, 103, 104, 105, 106, 107, 108] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00 42.00 43.00 44.00 | [10, 11, 12, 13, 14, 15, 16, 17, 18] | [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 1100 | [100, 101, 102, 103, 104, 105, 106, 107, 108, 109] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00 42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 | [10, 11, 12, 13, 14, 15, 16, 17, 18, 19] | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 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, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00 42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00 52.00 53.00 54.00 | [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 1130 | [100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00 42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00 52.00 53.00 54.00 55.00 56.00 57.00 58.00 59.00 60.00 61.00 62.00 63.00 64.00 | [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 1140 | [100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00 42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00 52.00 53.00 54.00 55.00 56.00 57.00 58.00 59.00 60.00 61.00 62.00 63.00 64.00 65.00 66.00 67.00 68.00 69.00 | [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23] | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total Within-Cluster Sum of Squares** | **Within-Cluster Sum of Squares** | **Cluster Centers** | **Cluster Sizes** | **Cluster Assignments for First 20 Observations** |
| 1150 | [100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114] | 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 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00 41.00 42.00 43.00 44.00 45.00 46.00 47.00 48.00 49.00 50.00 51.00 52.00 53.00 54.00 55.00 56.00 57.00 58.00 59.00 60.00 61.00 62.00 63.00 64.00 65.00 66.00 67.00 68.00 69.00 70.00 71.00 72.00 73.00 74.00 | [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24] | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 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"))) {

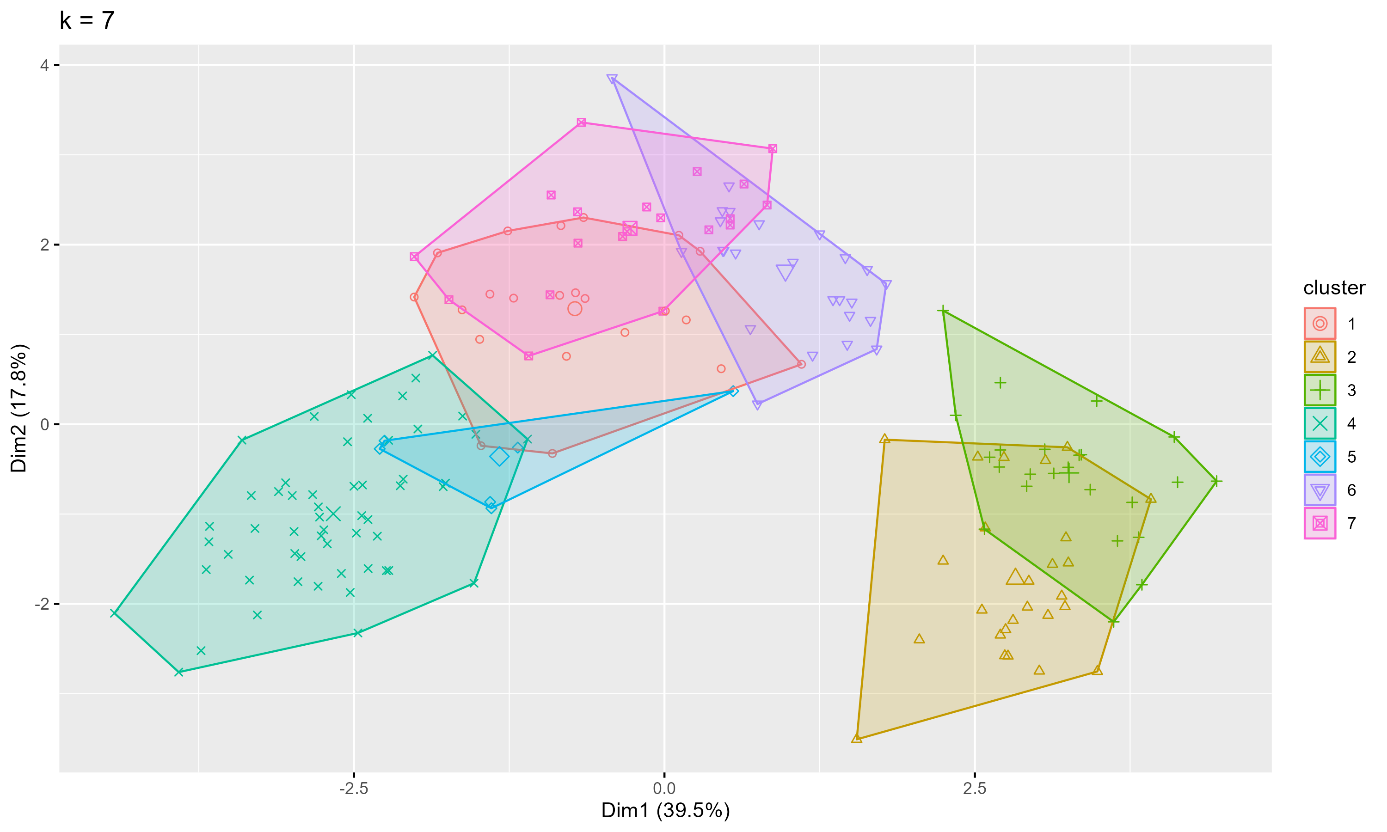
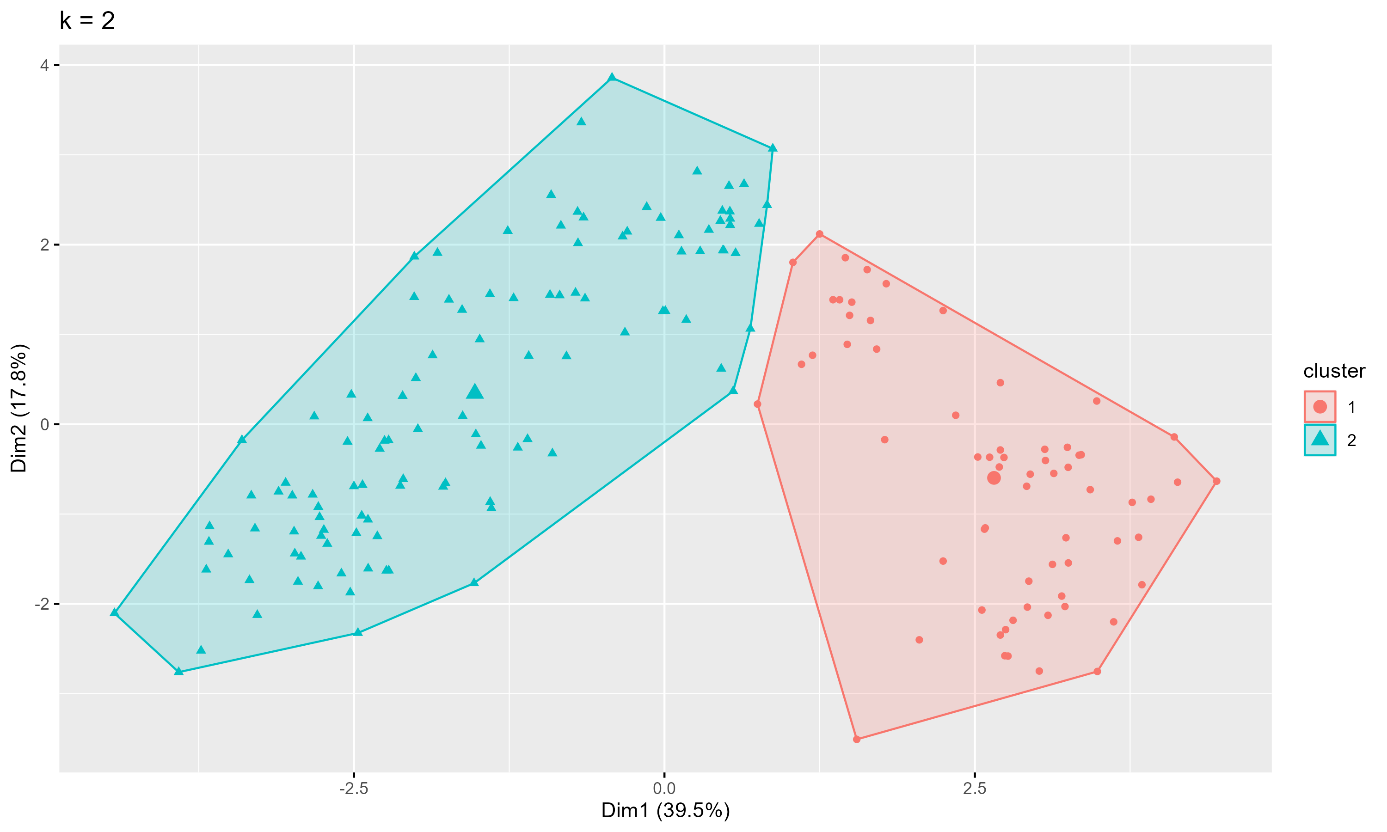
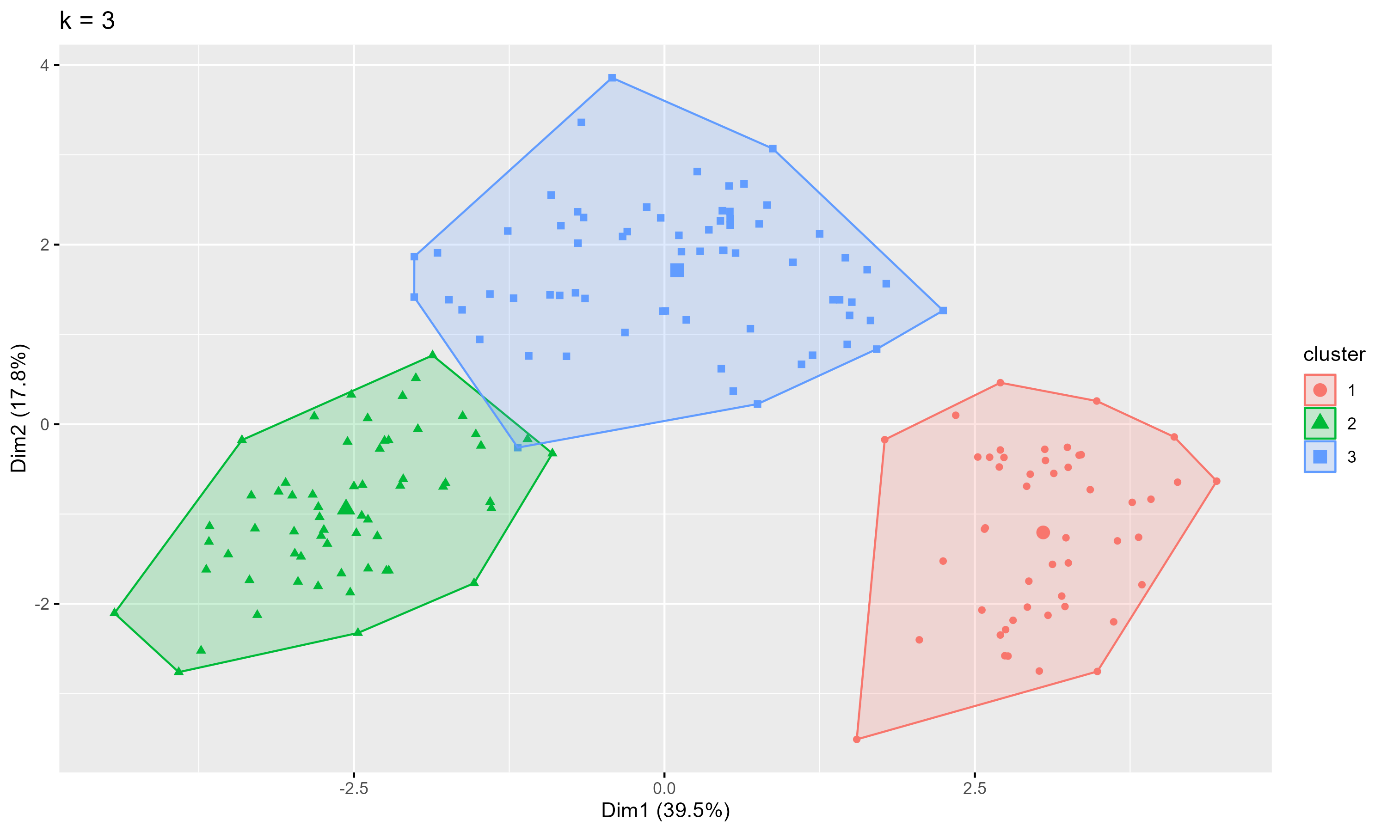
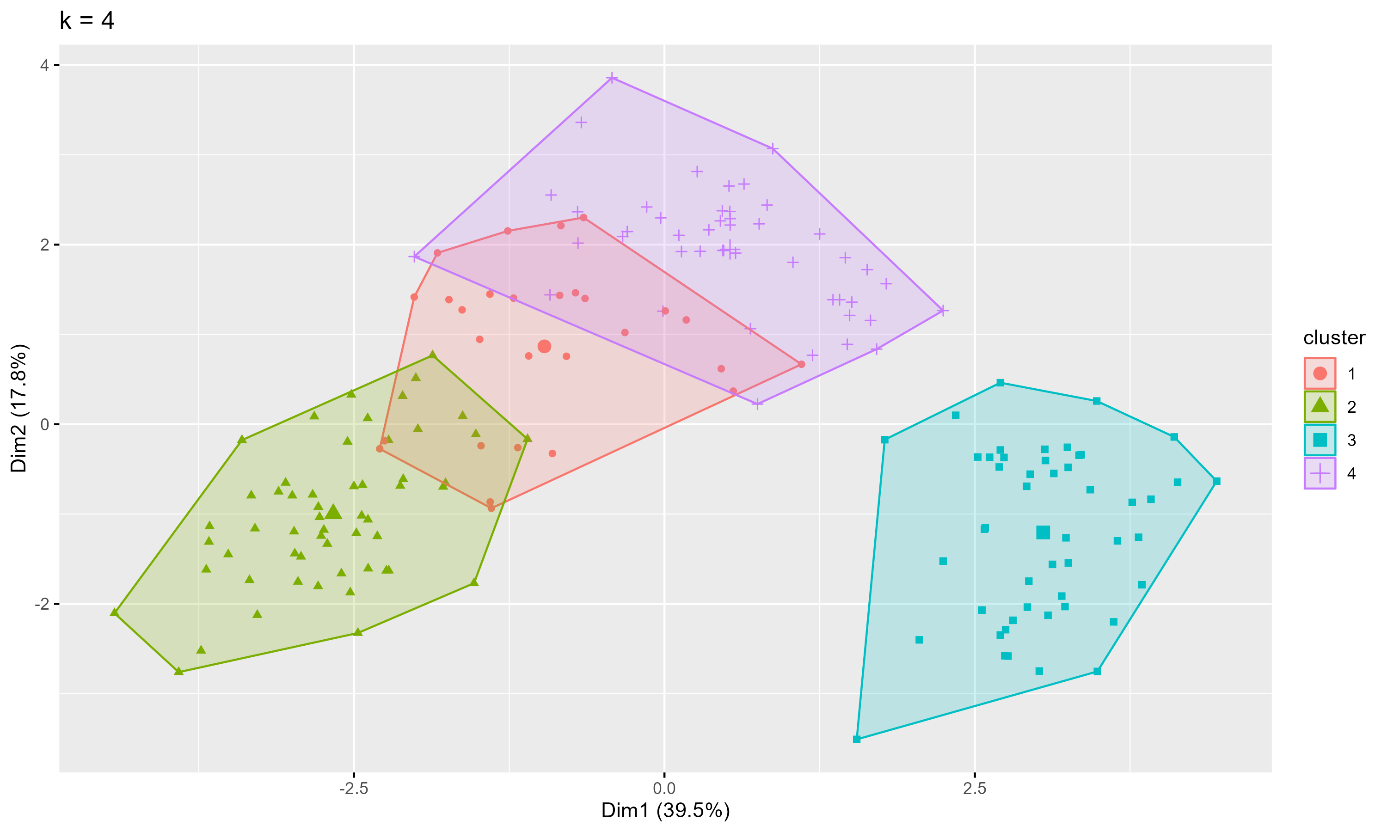
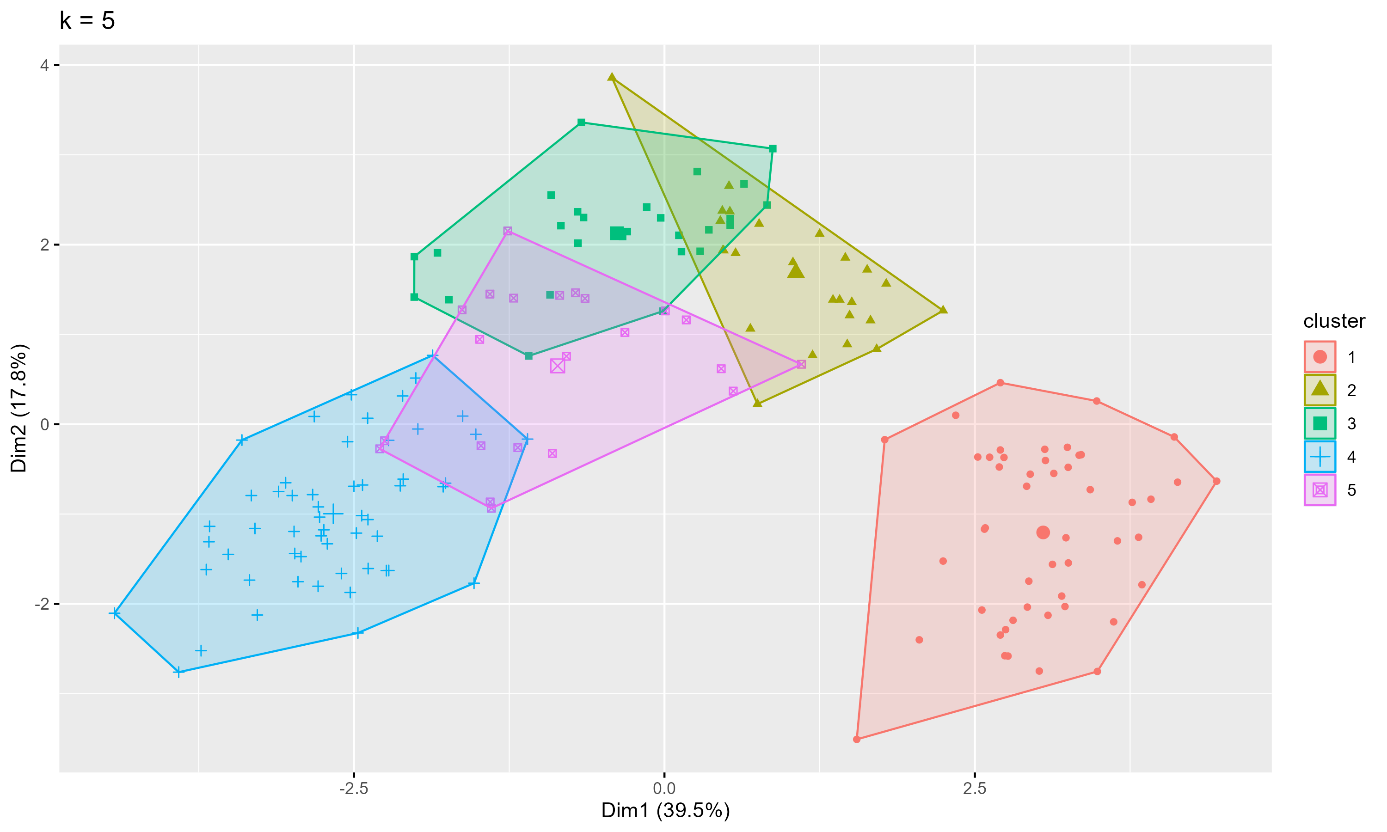
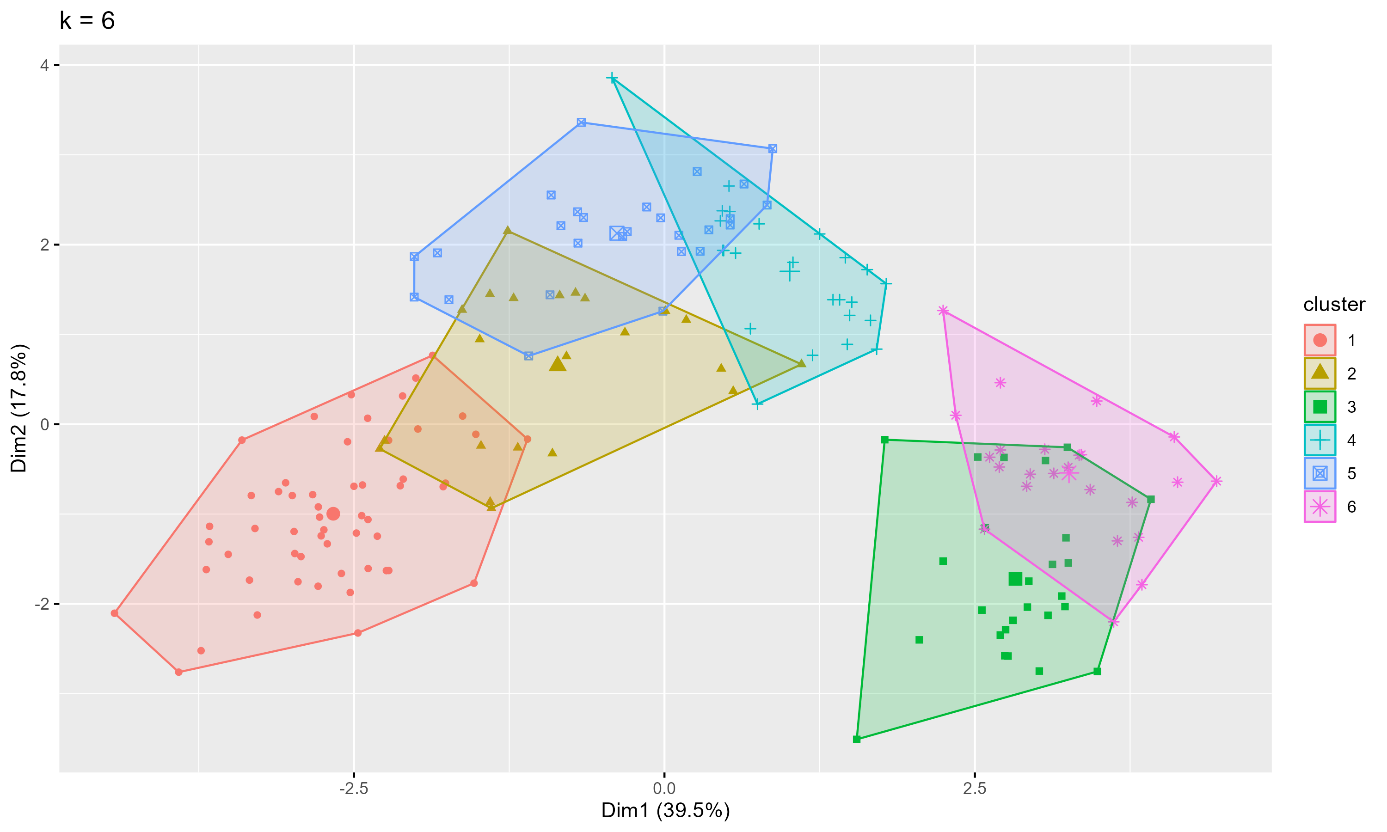
cat("Plik", paste("kmeans\_k", k, ".png"), "został zapisany.\n")

} 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")

Obraz zawierający tekst, linia, diagram, Wykres

Opis wygenerowany automatycznie

# # Metoda średniego konturu

fviz\_nbclust(wine\_data\_scaled, kmeans, method = "silhouette")

Obraz zawierający diagram, linia, tekst, Wykres

Opis wygenerowany automatycznie

# 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)

Obraz zawierający tekst, linia, diagram, Wykres

Opis wygenerowany automatycznie

## Wnioski

1. **Metoda łokcia (Elbow Method)**: Wykres pokazuje całkowitą sumę kwadratów wewnątrz klastrów dla różnych liczby klastrów k. Punkt, w którym krzywa zaczyna się wyginać (łokieć) i staje się mniej stroma, wskazuje na optymalną liczbę klastrów. 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 klastrów. Wartość ta mierzy, jak dobrze pasuje próbka do swojego klastra (spójność) w porównaniu do innych klastrów (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 klastrów z ich oczekiwanymi wartościami pod względem danych referencyjnych. Optymalną liczbę klastrów wskazuje największa luka. Na załączonym wykresie widoczny jest wyraźny szczyt dla k=3, co sugeruje, że najlepszą liczbą klastrów może być właśnie 3.

**Dyskusja i wybór optymalnej liczby klastrów**:

* 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

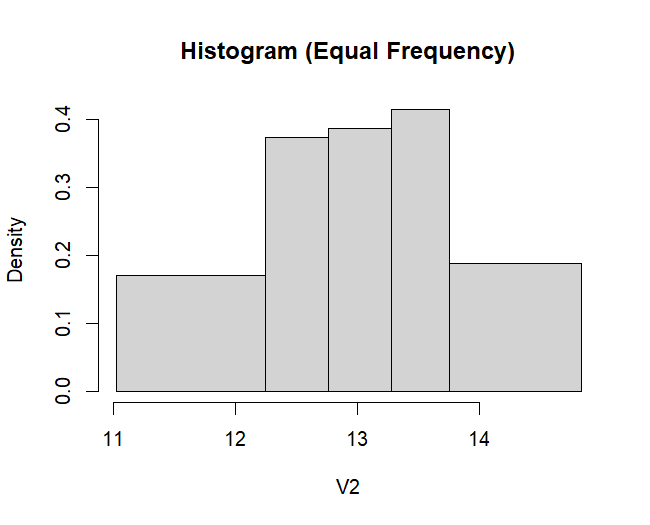
hist(variable\_to\_discretize, breaks = number\_of\_bins, main = "Histogram (Equal Width)", xlab = "V2")

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]

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

### 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"))

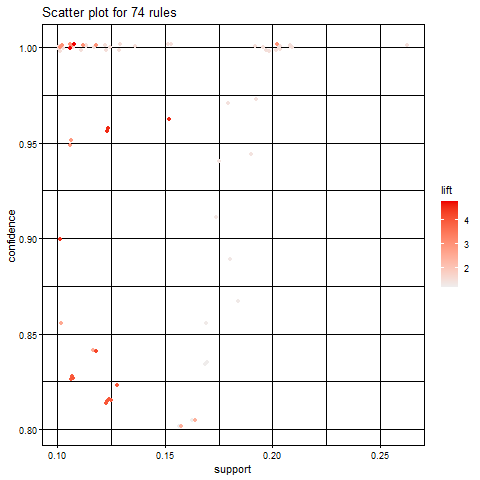
plot(rules, method = "scatter", measure = c("lift", "confidence"))

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

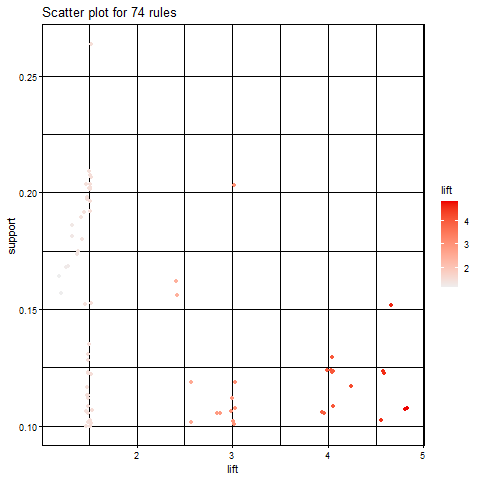
Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

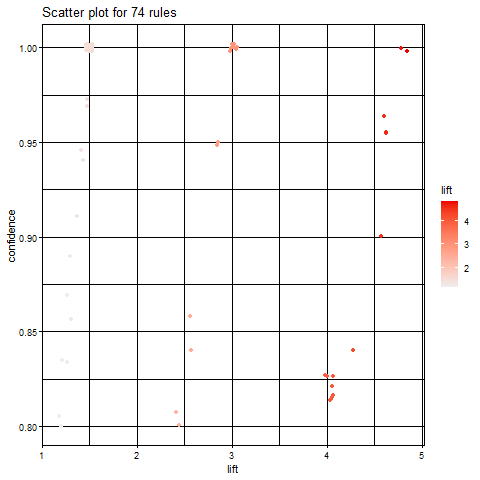
Opis wygenerowany automatycznie



Plot wykres\_wsparcie\_vs\_ufnosc



Plot wykres\_lift\_vs\_wsparcie



Plot 3 wykres\_lift\_vs\_ufnosc

# Lab 10

# Wczytanie potrzebnych bibliotek

library(e1071)

library(caret)

library(ggplot2)

# Ustawienie ścieżki dostępu do danych

path <- "C:/Users/petit/Desktop/repos/UO/rok 3/Wprowadzenie do eksploracji danych/lista10"

setwd(path)

# Wczytanie danych

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

# Podział danych na zbiór treningowy i testowy

set.seed(123) # Dla powtarzalności wyników

splitIndex <- createDataPartition(data[, ncol(data)], p = .8, list = FALSE)

train\_data <- data[splitIndex, ]

test\_data <- data[-splitIndex, ]