Dyskretyzacja i redukcja wymiaru na podstawie danych iris, City Quality of Life Dataset, titanic_train

Eksploracja danych

Tomasz Warzecha, album 282261

2025-04-24

Spis treści

1 Krótki opis zagadnienia

Tutaj umieszczamy:

- Co będziemy badali/analizowali?
- Na jakie pytania chcemy znaleźć odpowiedź?

2 Dyskretyzacja cech ciągłych

==========

Wczytanie danych

Tabela 1: Struktura danych

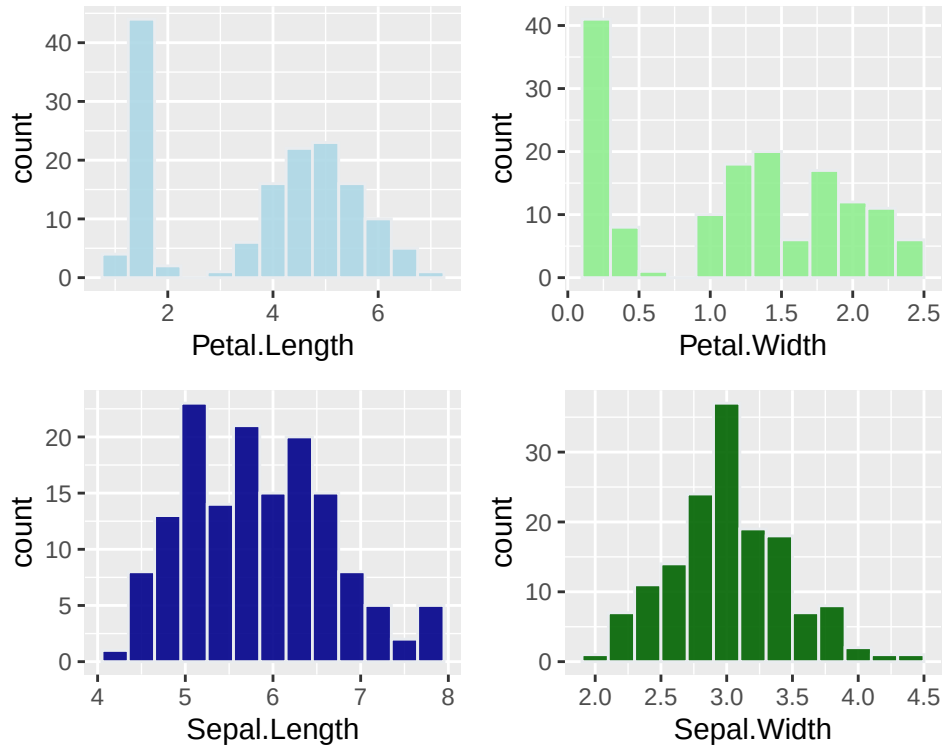
	x
Sepal.Length	numeric
Sepal.Width	numeric
Petal.Length	numeric
Petal.Width	numeric
Species	factor

Nasze dane mają 150 przypadków i 5 cech. W powyższej tabeli możemy zobaczyć wszystkie cechy oraz ich typy. Widzimy, że wszystkie zmienne zostały poprawnie rozpoznane. Nasze dane mają 4 cechy numeryczne oraz jedną jakościową. W żadnej z kolumn nie mamy braku wartości.

2.1 Wybór cech

Teraz przeanalizujemy nasze dane i wybierzemy zmienną o najgorszej i najlepszej zdolności dyskryminacyjnej.

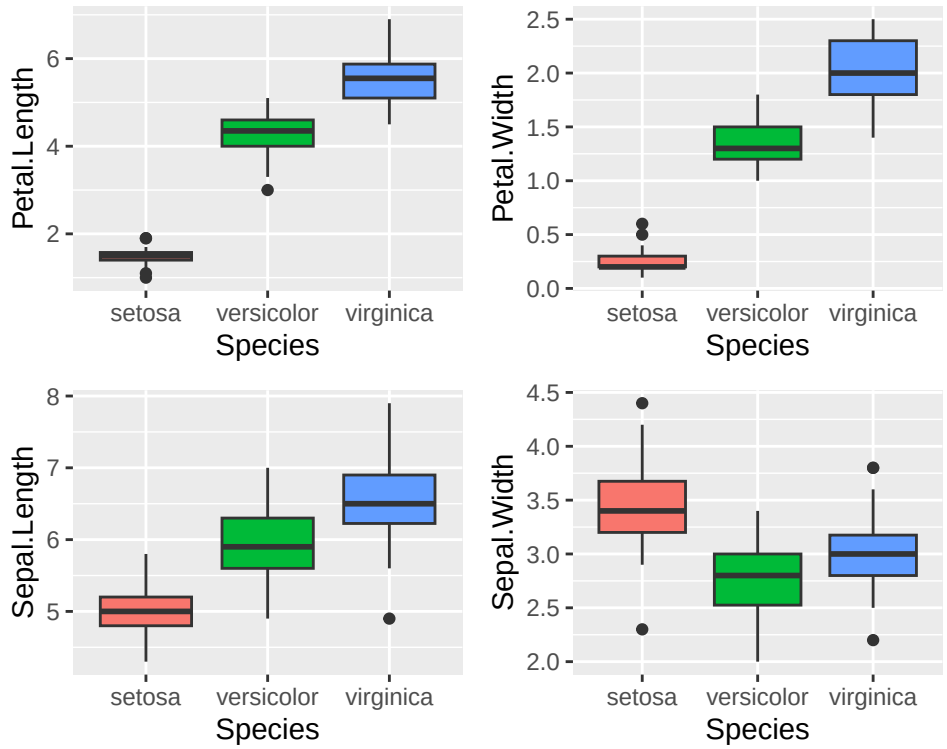
```
x <- iris[,"Petal.Length"]
n <- length(x)
y <- runif(n)
library(ggplot2)
library(gridExtra)
p1 <- ggplot(iris, aes(x = Petal.Length)) + geom_histogram( binwidth=0.5, fill="lightbl
p2 <- ggplot(iris, aes(x = Petal.Width)) + geom_histogram( binwidth=0.2, fill="lightgre
p3 <- ggplot(iris, aes(x = Sepal.Length)) + geom_histogram( binwidth=0.3, fill="darkblu
p4 <- ggplot(iris, aes(x = Sepal.Width)) + geom_histogram( binwidth=0.2, fill="darkgree
grid.arrange(p1,p2,p3,p4, nrow=2)
```



Najbardziej symetryczny wydaje sie rozklad zmiennej `Sepal.Width`, oraz mniej `Sepal.Length`. Zmienne opisujące płatek wyraznie wybijają dla małych warości, a następnie reszta wartości ma bardziej symetryczny rozkład. Może to sugerować np. mniejsze płatki kwiatów dla jednego z gatunków co może się nam przydać w dalszej analizie.

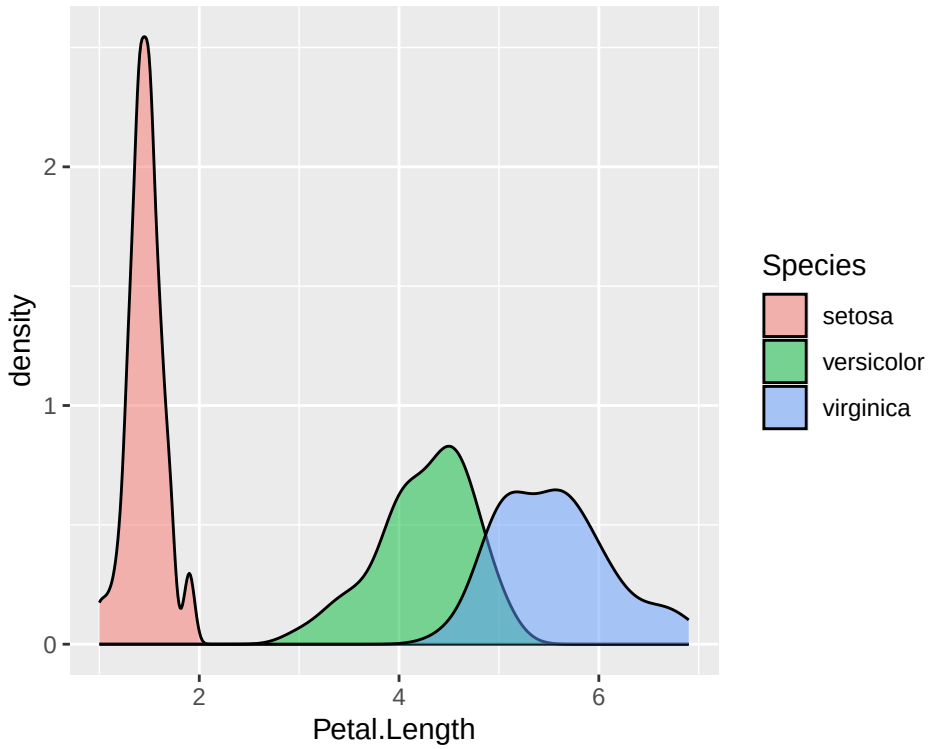
```
library(ggplot2)
library(gridExtra)
kolory <- c('lightblue','lightgreen','salmon')
p1 <- ggplot(iris, aes(x = Species, y = Petal.Length, fill = Species)) + geom_boxplot()
p2 <- ggplot(iris, aes(x = Species, y = Petal.Width, fill = Species)) + geom_boxplot()
p3 <- ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species)) + geom_boxplot()
p4 <- ggplot(iris, aes(x = Species, y = Sepal.Width, fill = Species)) + geom_boxplot()

grid.arrange(p1,p2,p3,p4, nrow = 2)
```

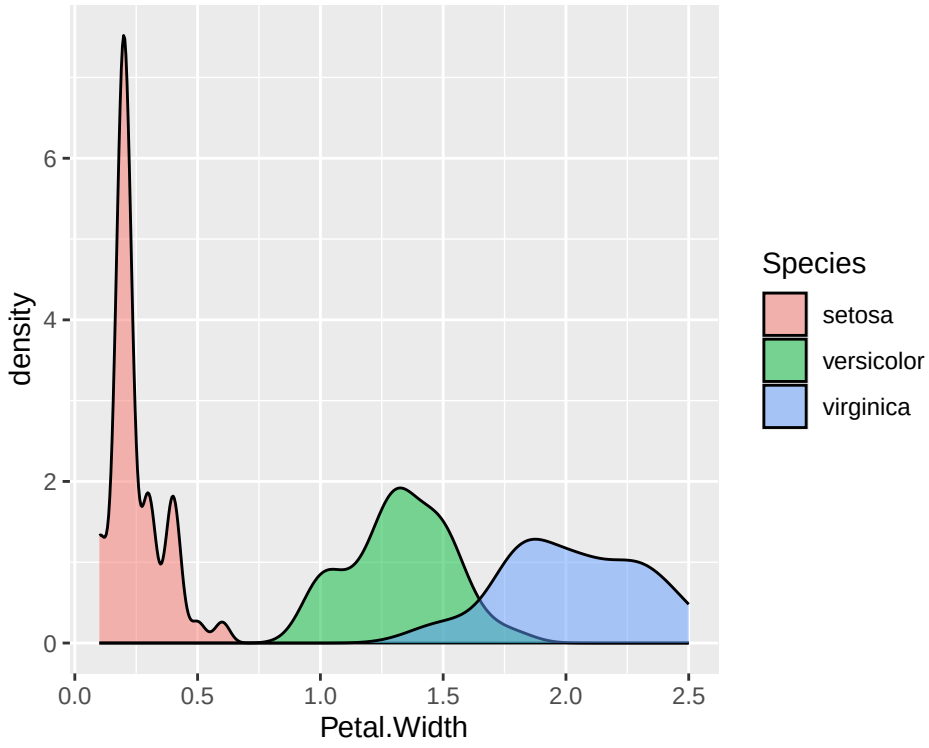


Na powyższych wykresach pudełkowych wyraźnie widać, że zmienne dotyczące dzwonka (Sepal), nie pozwolą nam dokładnie rozgraniczyć naszych gatunków. Zdecydowanie najgorszą cechą pod tym względem jest `Sepal.Width`, której praktycznie wszystkie trzy pudełka się nakładają. Zmienne dotyczące wymiarów płatka zdecydowanie lepiej pozwolą nam zidentyfikować gatunki, natomiast przyjżyjmy się im lepiej, aby wybrać najlepszą cechę.

```
library(ggplot2)
ggplot(iris, aes(x = Petal.Length, fill = Species)) +
  geom_density(alpha = 0.5)
```




```
library(ggplot2)
ggplot(iris, aes(x = Petal.Width, fill = Species)) +
  geom_density(alpha = 0.5)
```



Na powyższych wykresach rozkładu widzimy, że gatunek setosa jest dobrze rozróżnialny,

natomiast versicolor i virginica delikatnie się nakładają w podobnym stopniu dla długości i szerokości (jednak tu delikatnie mniej). Natomiast przez fakt, że zmienna `Petal.Length` będzie miała wyższe statystyki, a zatem większe różnice między grupami, wybieramy tą zmienną jako najlepszą do dyskryminacji naszych danych.

2.2 Porównanie nienadzorowanych metod dyskretyzacji

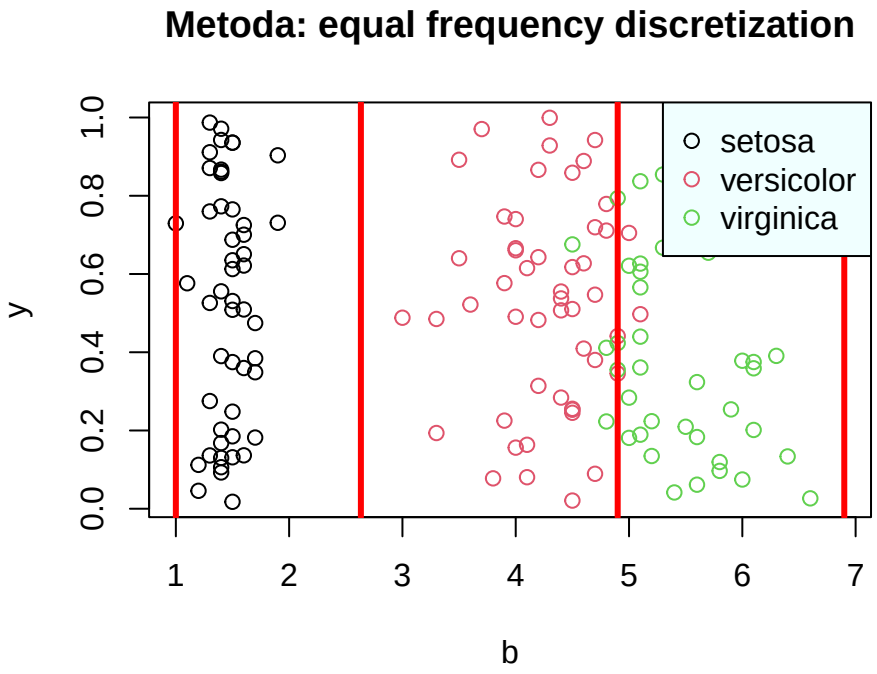
```
library(arules)
library(cluster)
library(dplyr)
b <- iris[,"Petal.Length"]
w <- iris[,"Sepal.Width"]
n <- length(b)
y = runif(n)
b.disc.equal.freq <- discretize(b, breaks = 3)
t1 <-table(b.disc.equal.freq, iris$Species)
t1
```

```
##
## b.disc.equal.freq setosa versicolor virginica
##         [1,2.63)       50          0         0
##         [2.63,4.9)      0         46         3
##         [4.9,6.9]       0          4        47
```

```
w.disc.equal.freq <- discretize(w, breaks = 3)
t2 <- table(w.disc.equal.freq, iris$Species)
```

Powyższa tabela przedstawia nam wyniki dysktretyzacji opartej na równych częstościach

```
breaks.equal.frequency <- attributes(b.disc.equal.freq)$"discretized:breaks"
plot(b, y, col=iris$Species, main = "Metoda: equal frequency discretization")
abline(v = breaks.equal.frequency, col = "red", lwd=3)
legend(x = "topright", legend=levels(iris$Species), col=1:3, pch=21, bg = "azure")
```



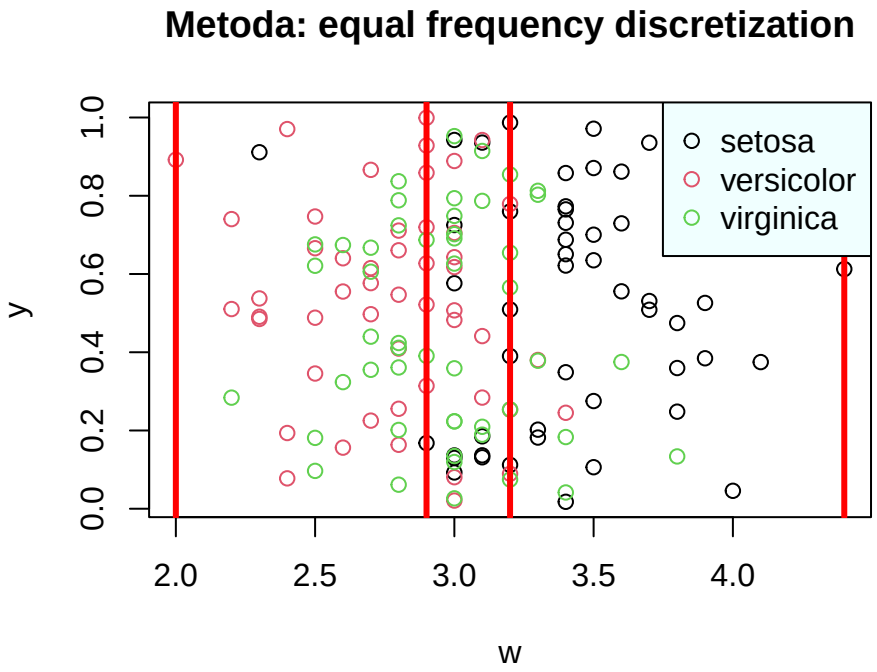

```
library(e1071)
matchClasses(t1)
```

```
## Cases in matched pairs: 95.33 %
```

```
##     [1,2.63)   [2.63,4.9)    [4.9,6.9]
##     "setosa" "versicolor"  "virginica"
```

Widzimy że metoda equal frequency discretization poradziła sobie dostyć dobrze, ze zgodnością ok. 95,3% dla zmiennej `Petal.Length`. Zobaczmy teraz jak to wygląda dla zmiennej `Sepal.Width`

```
breaks.equal.frequency <- attributes(w.disc.equal.freq)$"discretized:breaks"
plot(w, y, col=iris$Species, main = "Metoda: equal frequency discretization")
abline(v = breaks.equal.frequency, col = "red", lwd=3)
legend(x = "topright", legend=levels(iris$Species), col=1:3, pch=21, bg = "azure")
```




```
library(e1071)
matchClasses(t2)
```

```
## Cases in matched pairs: 55.33 %
```

```
##      [2,2.9)    [2.9,3.2)    [3.2,4.4]
## "versicolor" "versicolor"     "setosa"
```

Widzimy, że dla tej zmiennej kompletnie zawodzi dyskretyzacja. Dopasowanie jest na poziomie ok. 55,3%.

Przeprowadźmy teraz analize dla dyskretyzacji opartej na przedziałach o jednakowej szerokości (ang. equal interval width)

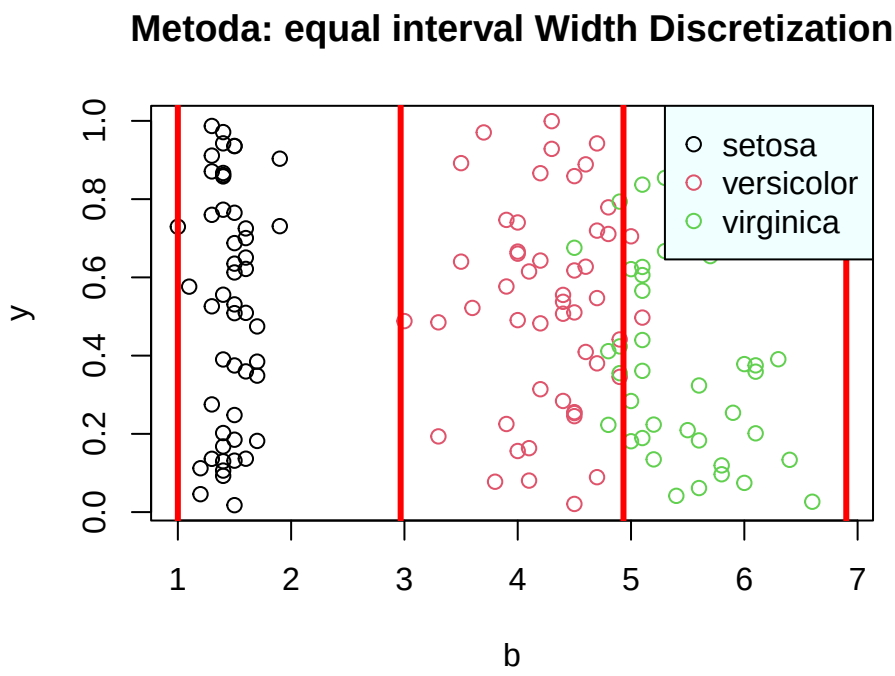
```
b.disc.equal.width <- discretize(b, method = "interval", breaks = 3)
t1 <-table(b.disc.equal.width, iris$Species)
t1
```

```
##
## b.disc.equal.width setosa versicolor virginica
##        [1,2.97)        50          0         0
##        [2.97,4.93)      0         48         6
##        [4.93,6.9]       0          2        44
```

```
w.disc.equal.width <- discretize(w, method = "interval", breaks = 3)
t2 <- table(w.disc.equal.width, iris$Species)
```

Powyższa tabela przedstawia nam wyniki dysktretyzacji opartej na przedziałach o jednakowej szerokości

```
breaks.equal.width <- attributes(b.disc.equal.width)$"discretized:breaks"
plot(b, y, col=iris$Species, main = "Metoda: equal interval Width Discretization")
abline(v = breaks.equal.width, col = "red", lwd=3)
legend(x = "topright", legend=levels(iris$Species), col=1:3, pch=21, bg = "azure")
```




```
matchClasses(t1)
```

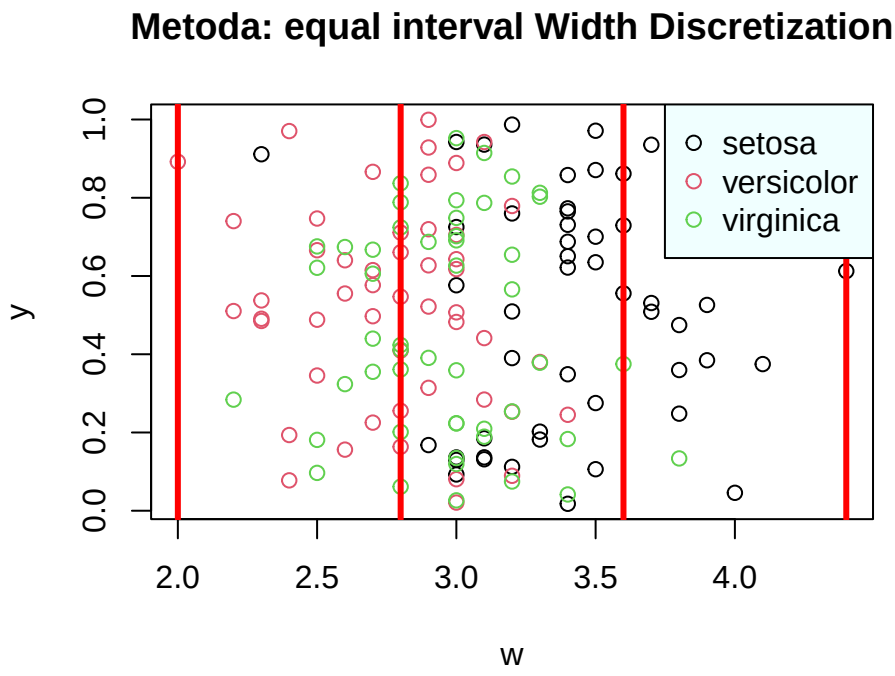
```
## Cases in matched pairs: 94.67 %
```

```
##     [1,2.97)  [2.97,4.93)   [4.93,6.9]
##     "setosa" "versicolor"  "virginica"
```

Widzimy, że ta metoda jest również bardzo dobra, ma zgodność na poziomie ok. 94,7%.

Sprawdźmy teraz jak wygląda ta metoda dla zmiennej `Sepal.Width`:

```
breaks.equal.width <- attributes(w.disc.equal.width)$"discretized:breaks"
plot(w, y, col=iris$Species, main = "Metoda: equal interval Width Discretization")
abline(v = breaks.equal.width, col = "red", lwd=3)
legend(x = "topright", legend=levels(iris$Species), col=1:3, pch=21, bg = "azure")
```




```
matchClasses(t2)
```

```
## Cases in matched pairs: 50.67 %
```

```
##      [2,2.8)    [2.8,3.6)    [3.6,4.4]
## "versicolor"     "setosa"     "setosa"
```

Dla tej zmiennej również zgodność jest słaba, wynosi ok. 50,7%

Sprawźmy teraz jak wygląda dyskretyzacja oparta na algorytmie grupowania (ang. k-means discretization)

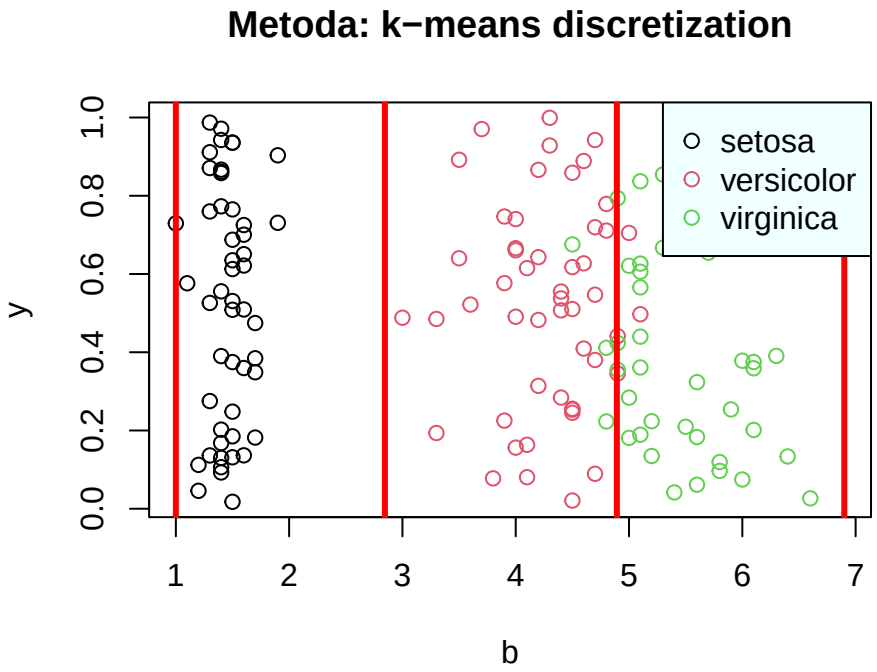
```
b.disc.k.means <- discretize(b, method = "cluster", breaks = 3)
t1 <-table(b.disc.k.means, iris$Species)
t1
```

```
##
## b.disc.k.means setosa versicolor virginica
##    [1,2.85)        50          0         0
##    [2.85,4.89)      0         46         3
##    [4.89,6.9]       0          4        47
```

```
w.disc.k.means <- discretize(w, method = "cluster", breaks = 3)
t2 <- table(w.disc.k.means, iris$Species)
```

Powyższa tabela przedstawia nam wyniki dysktretyzacji opartej na algorytmie grupowania.

```
breaks.k.means <- attributes(b.disc.k.means)$"discretized:breaks"
plot(b, y, col=iris$Species, main = "Metoda: k-means discretization")
abline(v = breaks.k.means, col = "red", lwd=3)
legend(x = "topright", legend=levels(iris$Species), col=1:3, pch=21, bg = "azure")
```




```
matchClasses(t1)
```

```
## Cases in matched pairs: 95.33 %
```

```
##     [1,2.85)  [2.85,4.89)   [4.89,6.9]
##     "setosa" "versicolor"  "virginica"
```

3 PCA - analiza składowych głównych

==========

4 MSD - skalowanie wielowymiarowe

==========

4.1 Podsumowanie

Najważniejsze wnioski, jakie udało się wysnuć na podstawie przeprowadzonych analiz/eksperymentów. Wnioski mogą być wypunktowane, tzn.:

- Tutaj wniosek nr 1
- Tutaj wniosek nr 2
- .....