Raport 2

Eksploracja danych

Mikołaj Langner, Marcin Kostrzewa
nr albumów: 255716, 255749

2021-04-19

Spis treści

1 Wstęp

Sprawozdanie zawiera rozwiązanie zadań z listy 2. Dotyczą one zagadnień dyskretyzacji i redukcji wymiaru.

2 Zadanie 1

W pierwszym zadaniu mamy dokonać dyskretyzacji cech ciągłych ze zbioru `iris` i ocenić jej jakość.

2.1 Wczytanie danych i wstępna analiza

```
data(iris)
```

Wybierzmy zmienne o najlepszej i najgorszej zdolności dyskryminacyjnej. W tym celu narysujemy wykresy pudełkowe oraz wyliczymy współczynniki zmienności każdej ze zmiennych z podziałem na poszczególne gatunki irysów i porównamy ich rozkłady.

```
plot_boxplot(iris, by="Species")
```

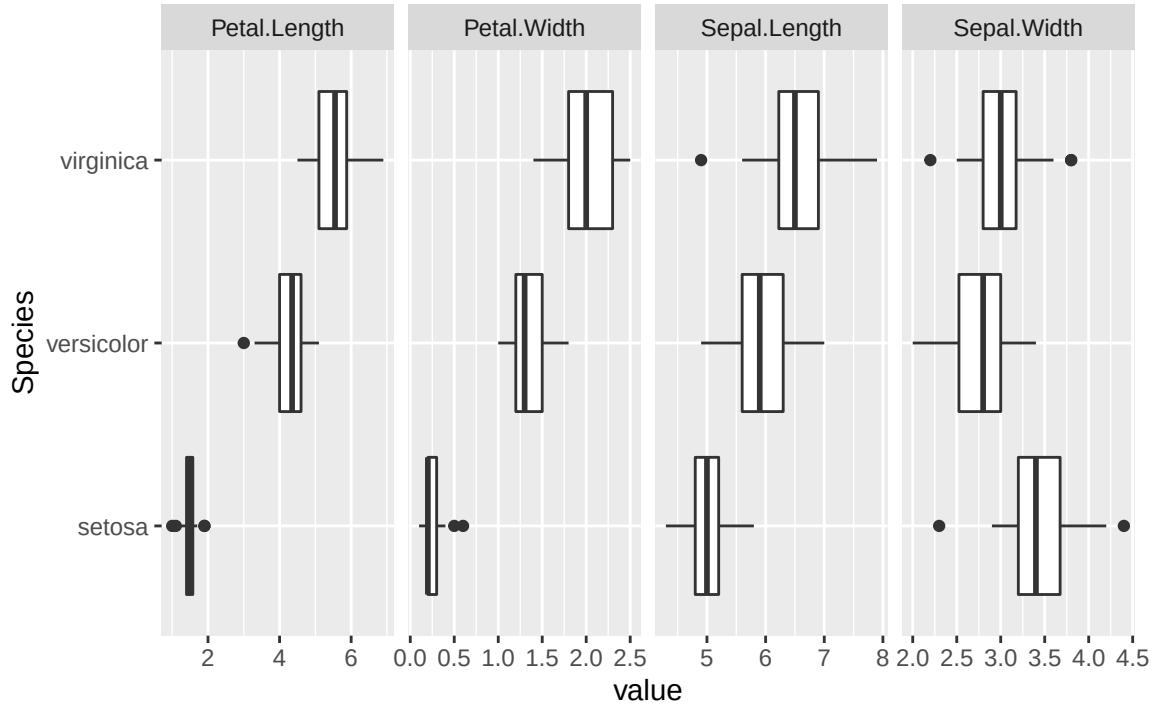


Tabela 1: Jakis label

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	0.070	0.111	0.119	0.428
versicolor	0.087	0.113	0.110	0.149
virginica	0.097	0.108	0.099	0.136

Możemy zauważyć, że zmienna Petal.Length najefektywniej rozdziela poszczególne gatunki, natomiast zmienna Sepal.Width radzi sobie z tym najgorzej.

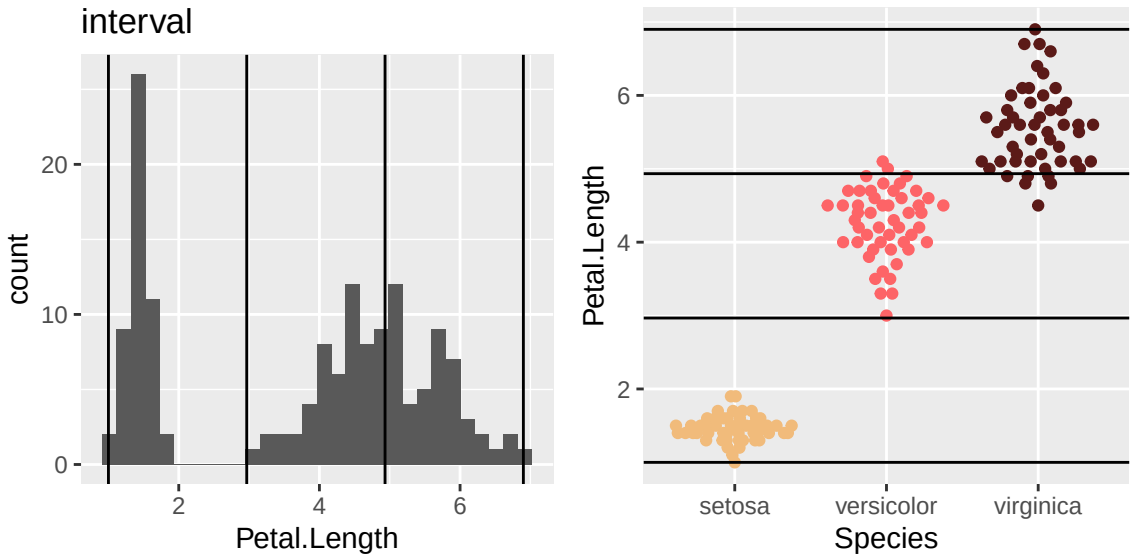
2.2 Metody dyskretyzacji

Porównamy ze sobą cztery metody dyskretyzacji nienadzorowanej:

- equal width,
- equal frequency,
- k-means clustering,
- dyskretyzację dla przedziałów zadanych przez użytkownika.

2.2.1 Najlepiej separująca zmienna

Zacznijmy od zmiennej Petal.Length, która najlepiej rozdziela poszczególne gatunki irysów.



Rysunek 1: 1

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

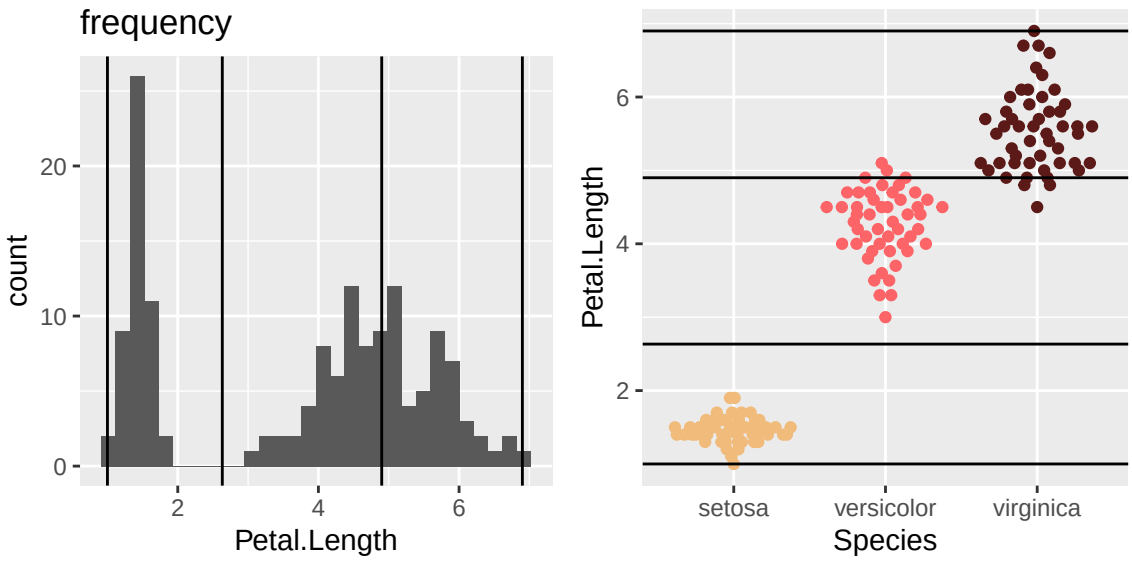
```
## Cases in matched pairs: 95.33 %
```

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

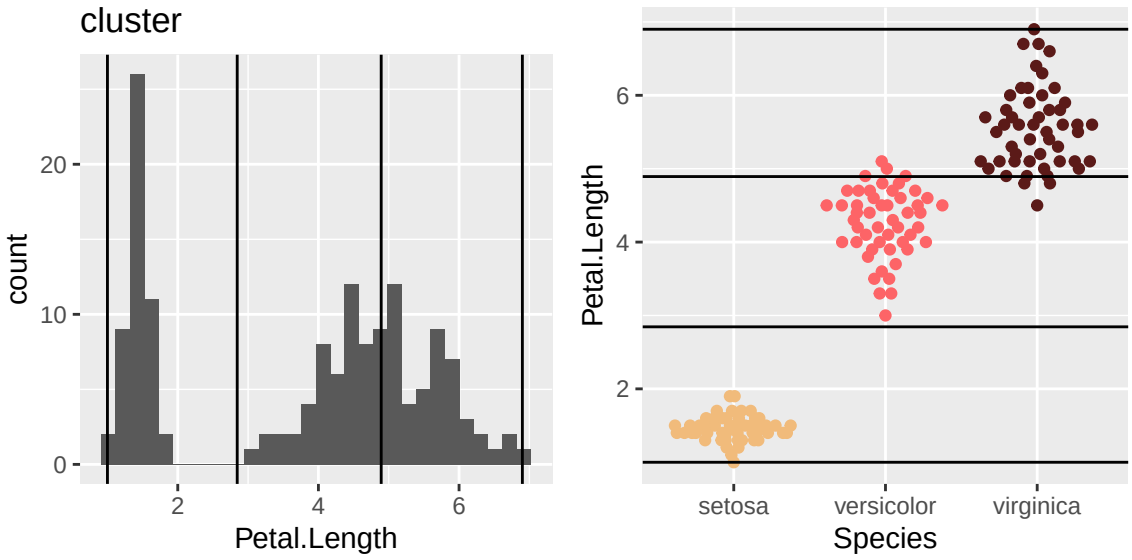
```
## Cases in matched pairs: 94.67 %
```

2.2.2 Najgorzej separująca zmienna

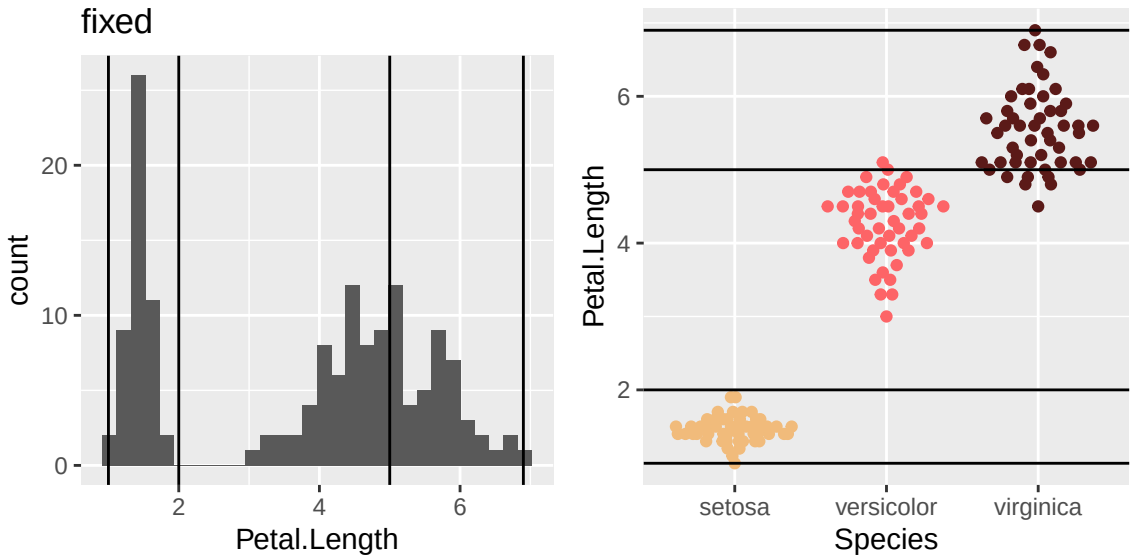
Możemy zobaczyć teraz jak poszczególne metody działają dla zmiennej Sepal.Width, która najgorzej radzi sobie z rozdzielaniem gatunków.



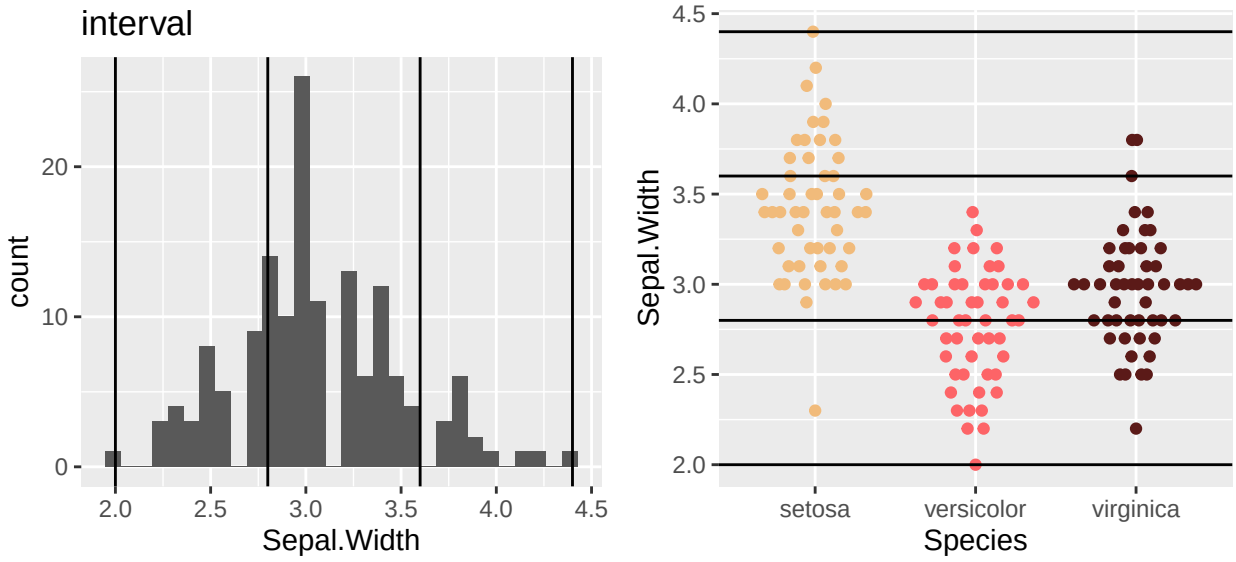
Rysunek 2: 2



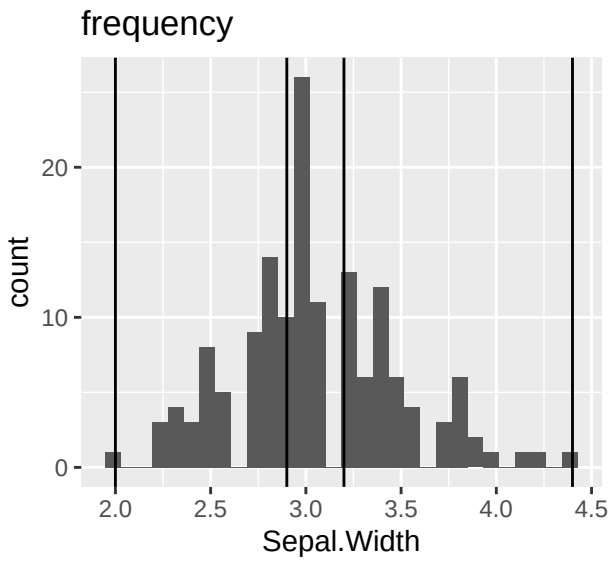
Rysunek 3: 3

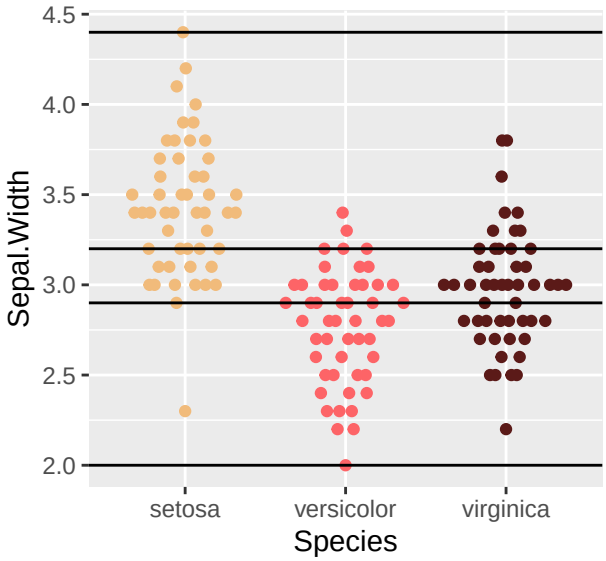


Rysunek 4: 4

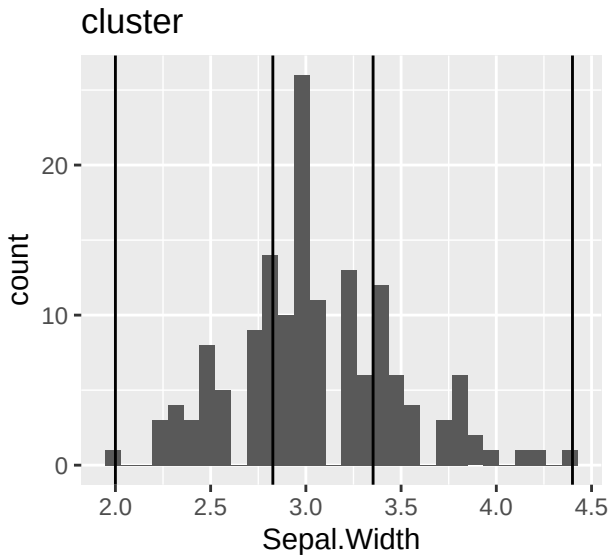


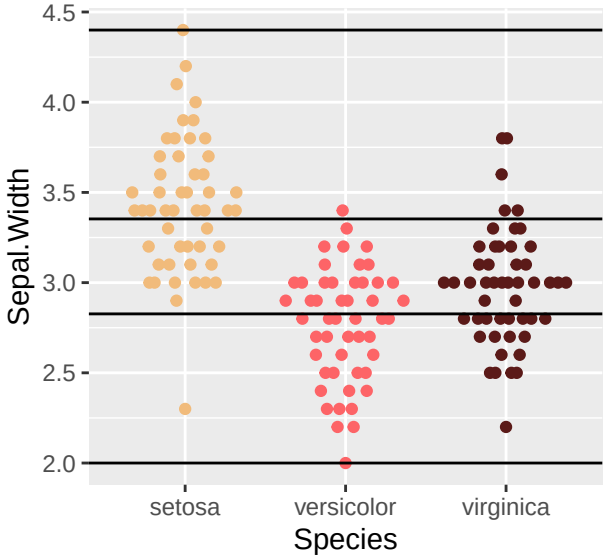

```
## Cases in matched pairs: 50.67 %
```



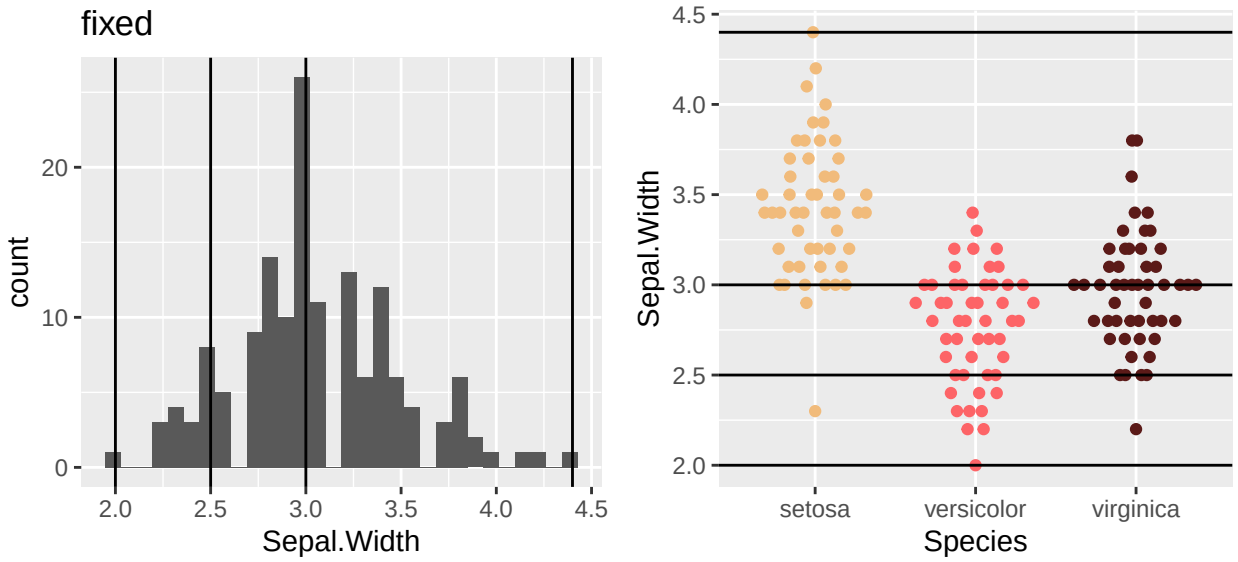



```
## Cases in matched pairs: 55.33 %
```






```
## Cases in matched pairs: 56 %
```




```
## Cases in matched pairs: 54.67 %
```

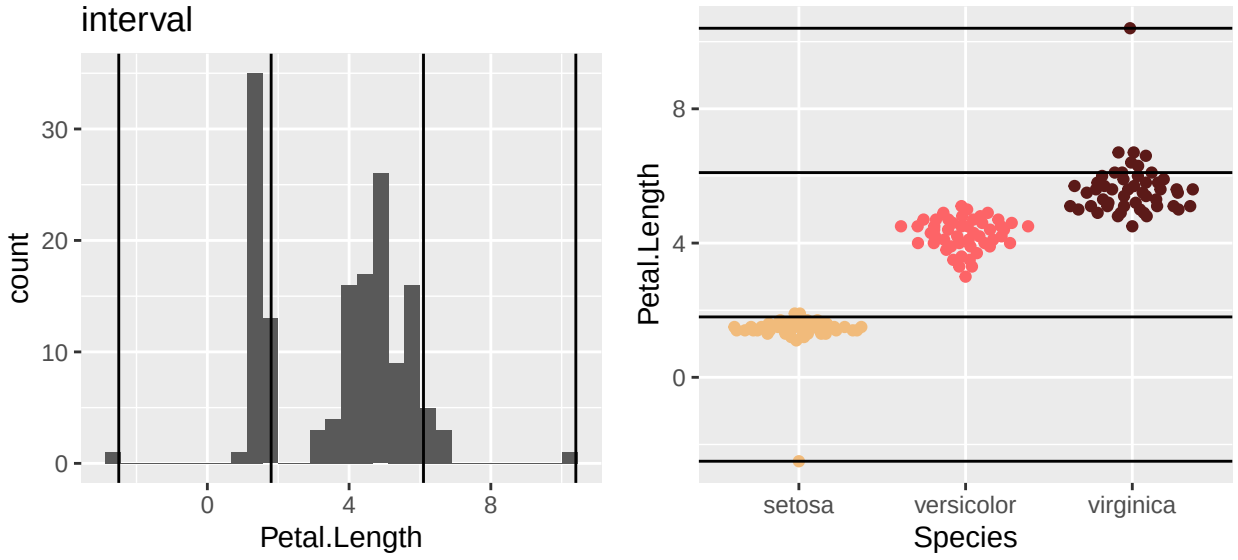
Dla obu zmiennych każda z metod wypada równie dobrze, przy czym, najlepsze wyniki produkują metody równej częstości oraz k-średnich.

2.3 Metody dyskretyzacji z wartościami odstającymi

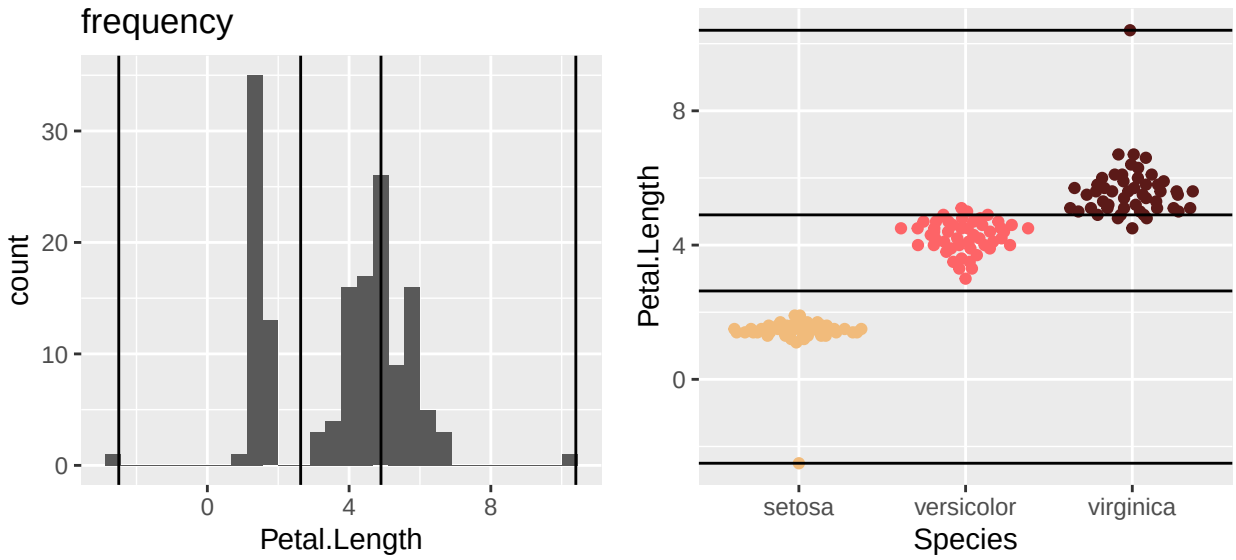
Rozpatrzmy teraz dyskretyzację przy dodaniu sztucznie wartości odstających.

2.3.1 Zmienna Petal.Length

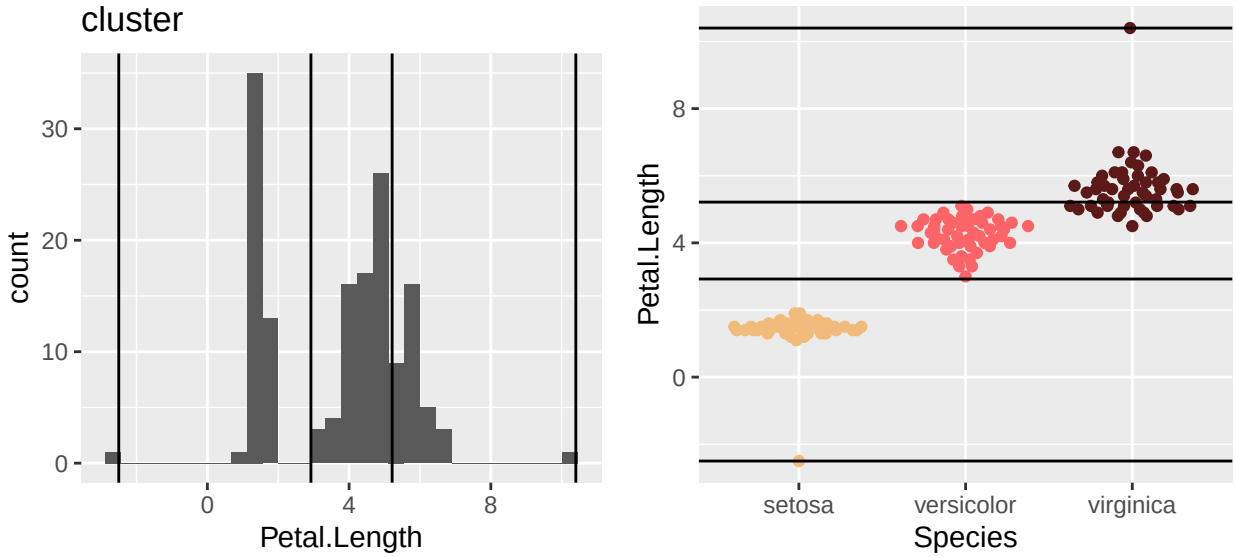
Zacznijmy znowu od zmiennej Petal.Length.



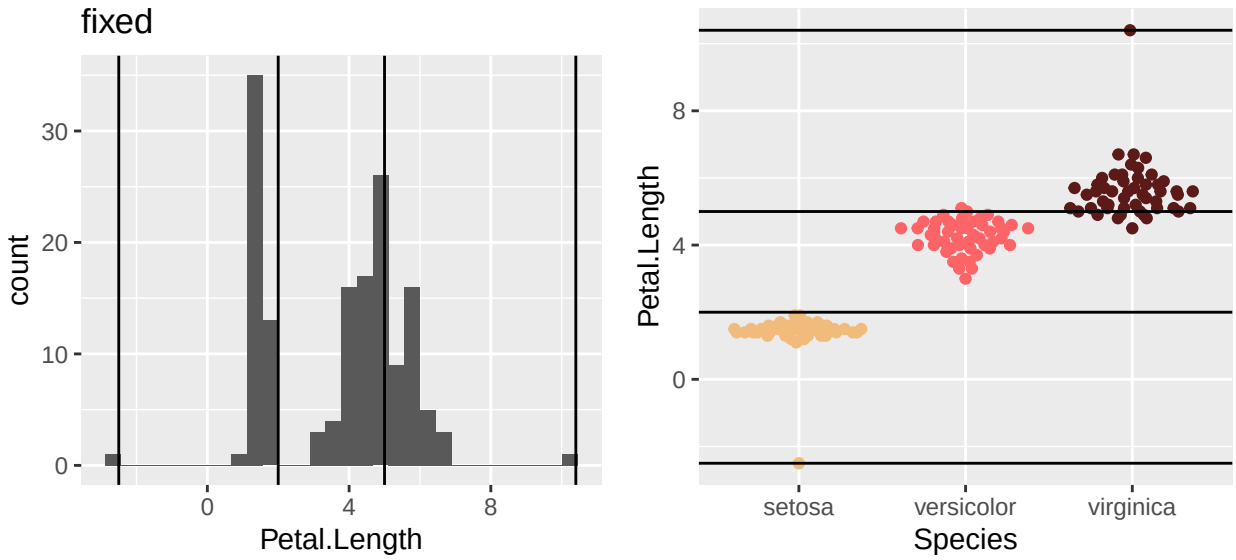

```
## Cases in matched pairs: 71.33 %
```




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



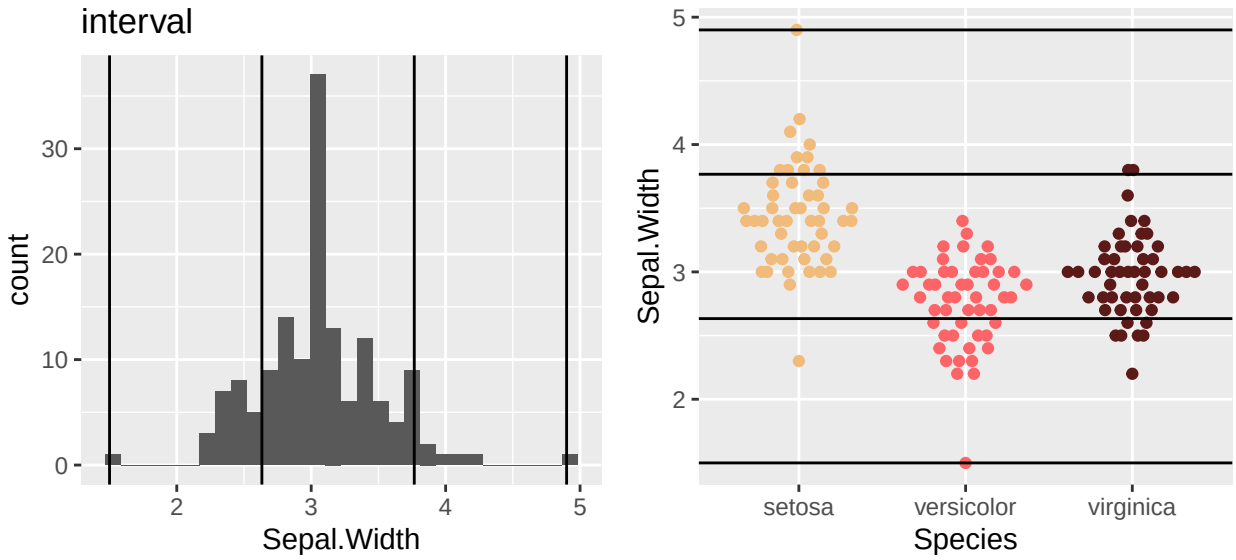

```
## Cases in matched pairs: 88 %
```



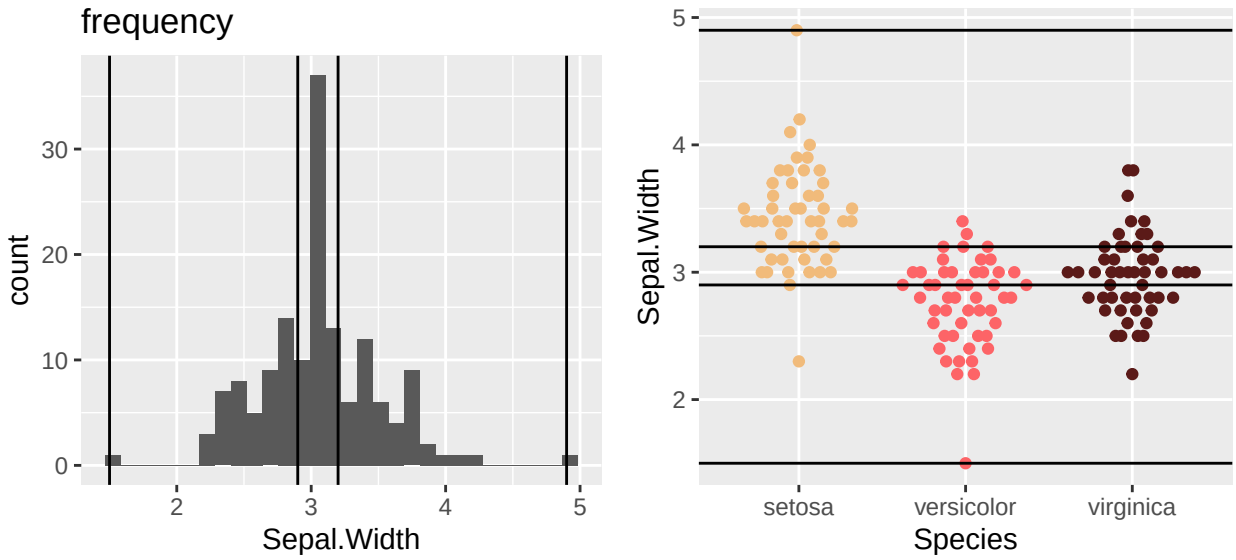

```
## Cases in matched pairs: 94.67 %
```

2.3.2 Zmienna Sepal.Width

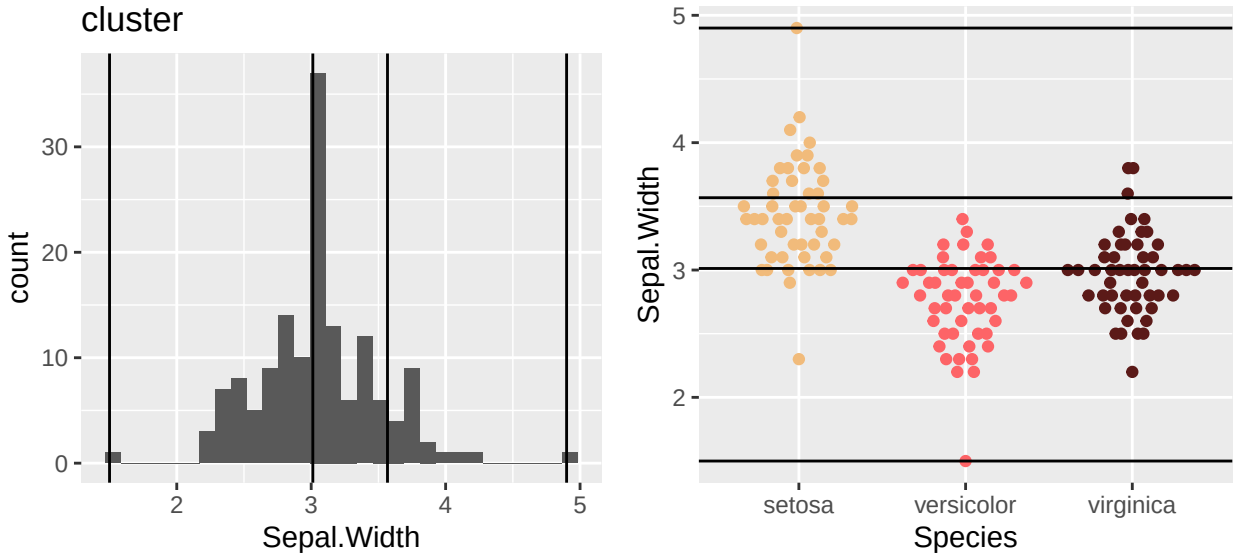
Dla zmiennej Sepal.Width po dodaniu wartości odstających dyskretyzacja wygląda następująco:



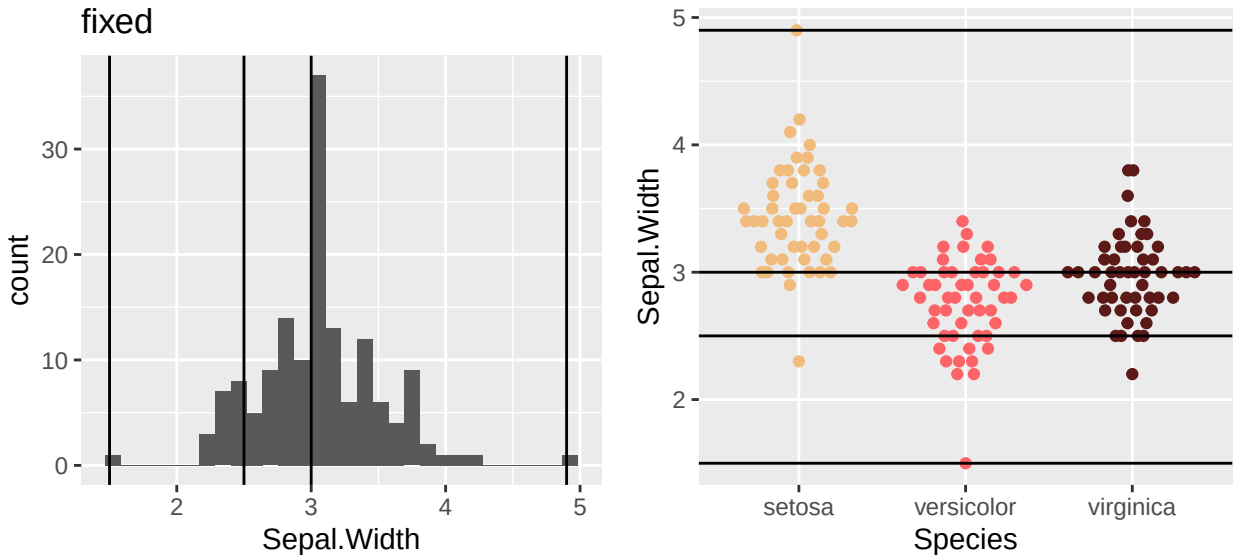

```
## Cases in matched pairs: 44.67 %
```




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




```
## Cases in matched pairs: 56 %
```




```
## Cases in matched pairs: 54.67 %
```

Nie powinien dziwić fakt, że największa zmiana w poprawności predykcji dotknęła metodę przedziałów równej długości, gdyż pojedyncza obserwacja całkowicie zmienia dobór miejsc partycji przedziału.

3 Zadanie 2

3.1 Wczytanie i przygotowanie danych

Teraz naszym zadaniem jest dokonanie analizy składów głównych (PCA) dla zbioru `state.x77`, który zbiera ...

Wczytajmy dane i uzupełnijmy je o informacje geograficzne o wszytkich stanach.

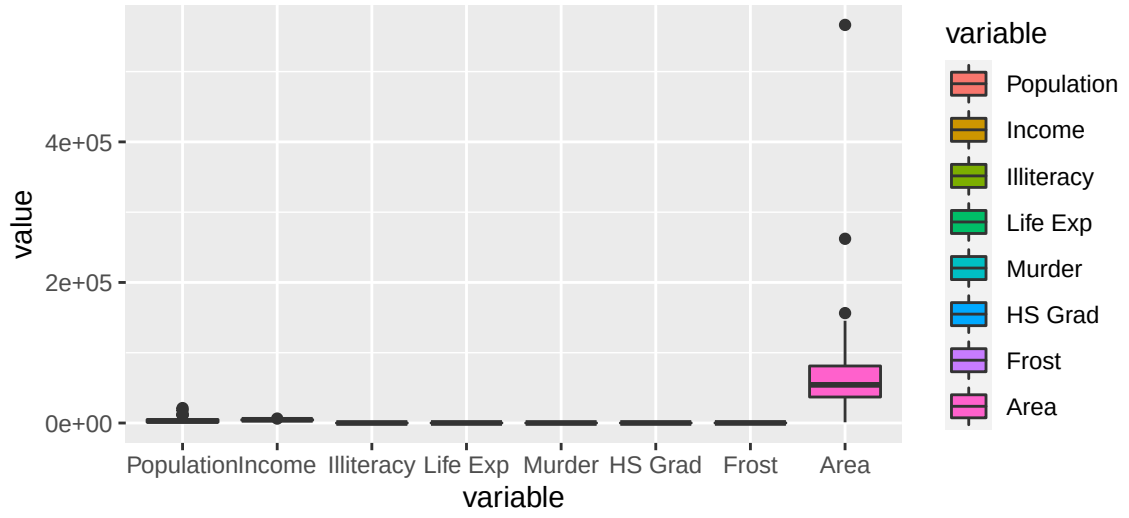
```
data(state)
state <- as.data.frame(state.x77)
state$region <- state.region
state$division <- state.division
state.subset <- subset(state, select=-c(region, division))
```

By rozstrzygnąć, czy potrzebna jest normalizacja danych, przeanalizujemy wykresy pudełkowe oraz wyznaczymy odchylenia standardowe i współczynniki zmienności.

Tabela 2: Odchylenie standardowe i wspolczynnik zmiennosci dla zmienych

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
Odchylenie standardowe	4464.491	614.470	0.610	1.342	3.692	8.077	51.981	85327.300
Wspolczynnik zmiennosci	1.051	0.139	0.521	0.019	0.500	0.152	0.498	1.206

Widać, że zmienne wymagają standaryzacji — ich wariancję zbyt mocno się różnią.

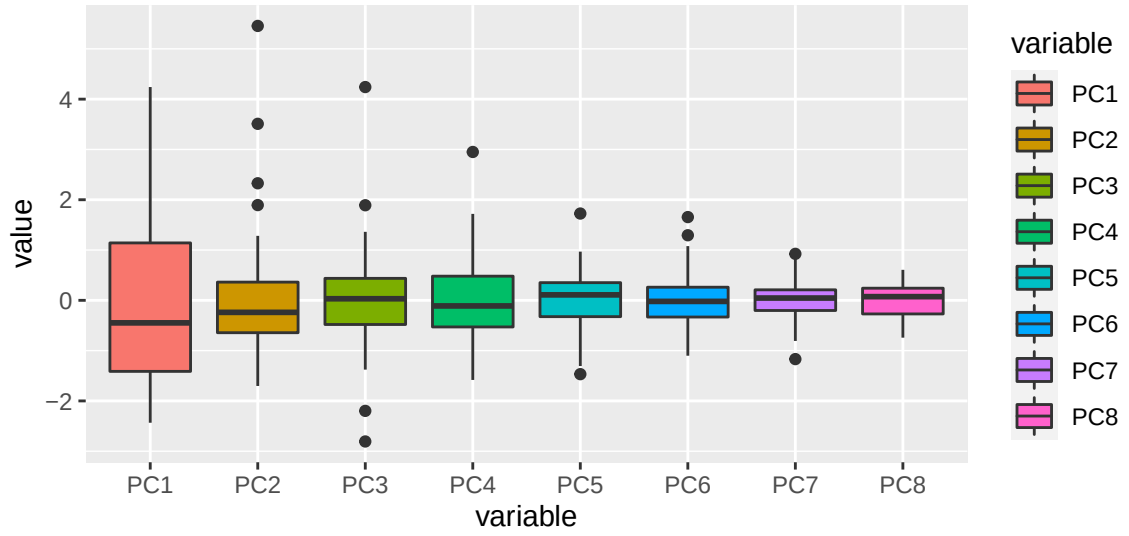


Rysunek 5: Wykresy pudelkowe dla zmiennych ze zbioru state.x77

3.2 Składowe główne i ich analiza

Wyznaczymy teraz składowe główne i przedstawimy ich rozrzut, wykorzystując wykresy pudełkowe.

```
after.pca <- prcomp(state.subset, retx=T, center=T, scale.=T)
```



Rysunek 6: Wykresy pudelkowe dla skladowych glownych

Przypatrzmy się teraz wektorom ładunków dla trzech pierwszych składowych głównych.

Wnioski — jeszcze się napisze ...

Zbadajmy teraz jaka część wyjaśnionej wariancji odpowiada kolejnym składowym głównym.

Tabela 3: Wektory ladunkow dla trzech pierwszych PC

	PC1	PC2	PC3
Population	0.126	0.411	-0.656
Income	-0.299	0.519	-0.100
Illiteracy	0.468	0.053	0.071
Life Exp	-0.412	-0.082	-0.360
Murder	0.444	0.307	0.108
HS Grad	-0.425	0.299	0.050
Frost	-0.357	-0.154	0.387
Area	-0.033	0.588	0.510

Tabela 4: Odchylenie standardowe i wspolczynnik zmiennosci dla zmienych

	PC1	PC2	PC3	PC4	PC5
Proporcja wariancji	0.45	0.204	0.139	0.088	0.048
Skumulowana wariancja	0.45	0.654	0.793	0.881	0.929

Zauważamy, że:

- PC1 wyjaśnia 45% wyjaśnianej wariancji, PC2 prawie 25%;
- 80% całkowitej wariancji jest wyjaśniane przez pierwsze cztery składowe główne (trzy pierwsze wyjaśniają niewiele mniej), 90% jest wyjaśniane zaś przez pierwszych 5.

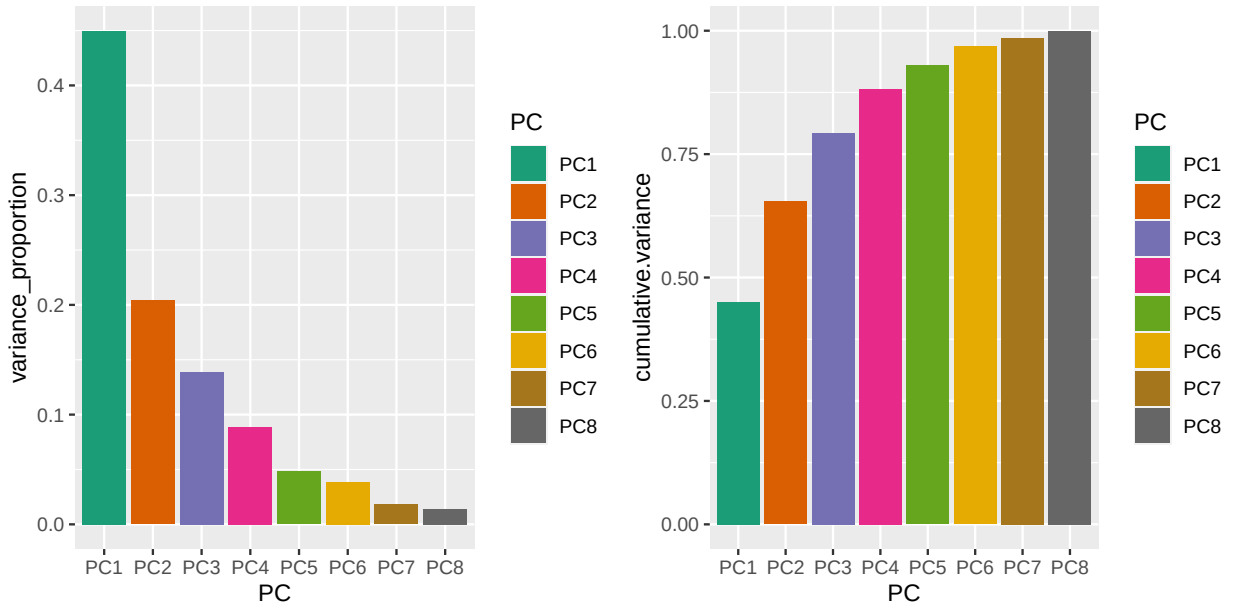
3.3 Wizualizacja danych

W tej części wygenerujemy wykresy rozrzutu 2d dla dwóch pierwszych składowych głównych.

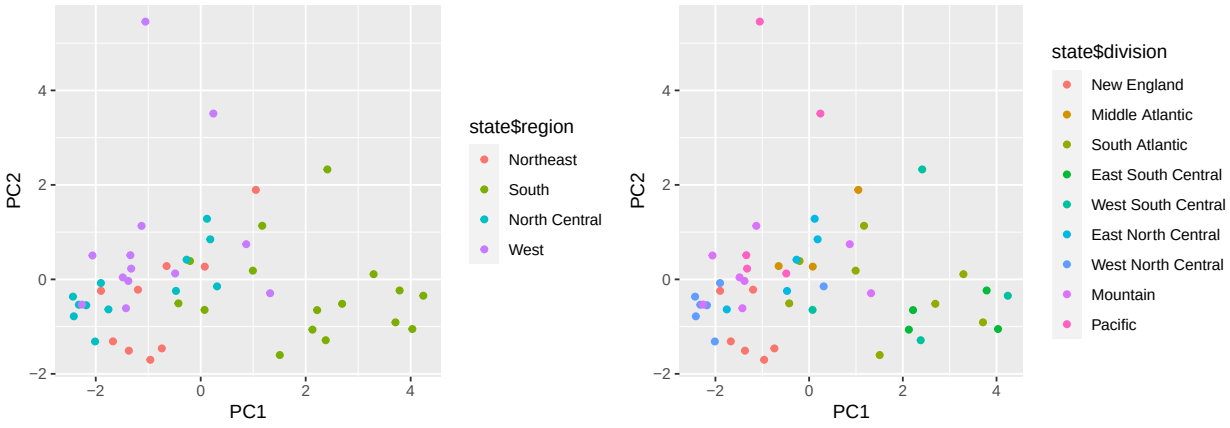
Przygotowaliśmy także wykresy 3d — kod umieściliśmy w dodatkowych skrypcie.

3.4 Korelacja zmiennych

Wygenerujmy teraz dwuwykres.



Rysunek 7: Wariancja wyjasniana przez poszczegolne skladowe glowne i wariancja skumulowana.



Rysunek 8: Wykresy rozrzutu

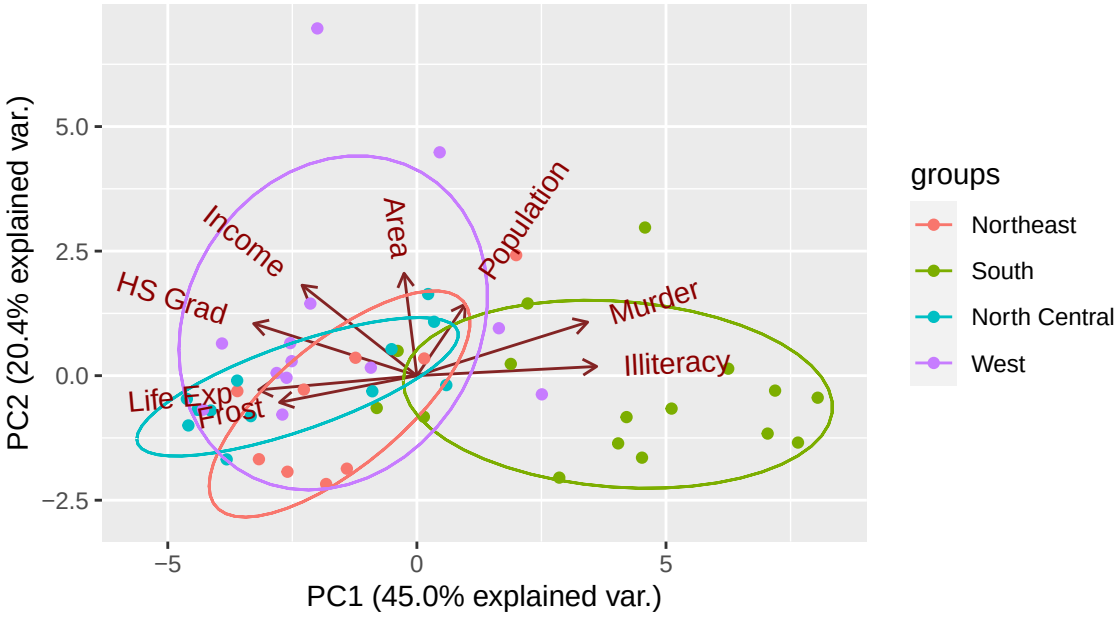
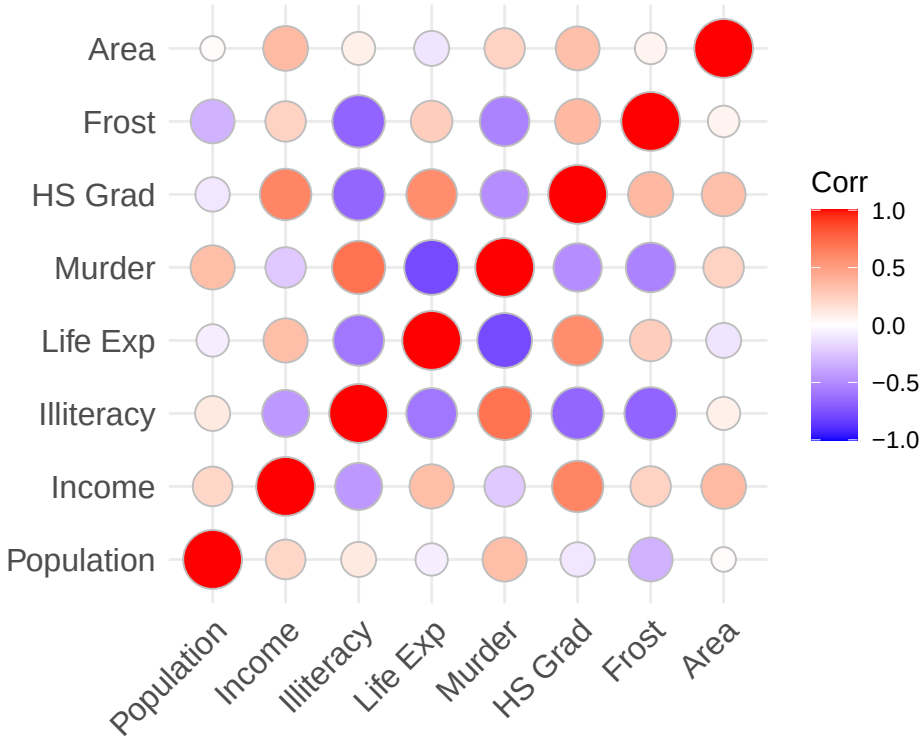


Tabela 5: Wartosci macierzy korelacji zmiennych

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
Population	1.000	0.208	0.108	-0.068	0.344	-0.098	-0.332	0.023
Income	0.208	1.000	-0.437	0.340	-0.230	0.620	0.226	0.363
Illiteracy	0.108	-0.437	1.000	-0.588	0.703	-0.657	-0.672	0.077
Life Exp	-0.068	0.340	-0.588	1.000	-0.781	0.582	0.262	-0.107
Murder	0.344	-0.230	0.703	-0.781	1.000	-0.488	-0.539	0.228
HS Grad	-0.098	0.620	-0.657	0.582	-0.488	1.000	0.367	0.334
Frost	-0.332	0.226	-0.672	0.262	-0.539	0.367	1.000	0.059
Area	0.023	0.363	0.077	-0.107	0.228	0.334	0.059	1.000



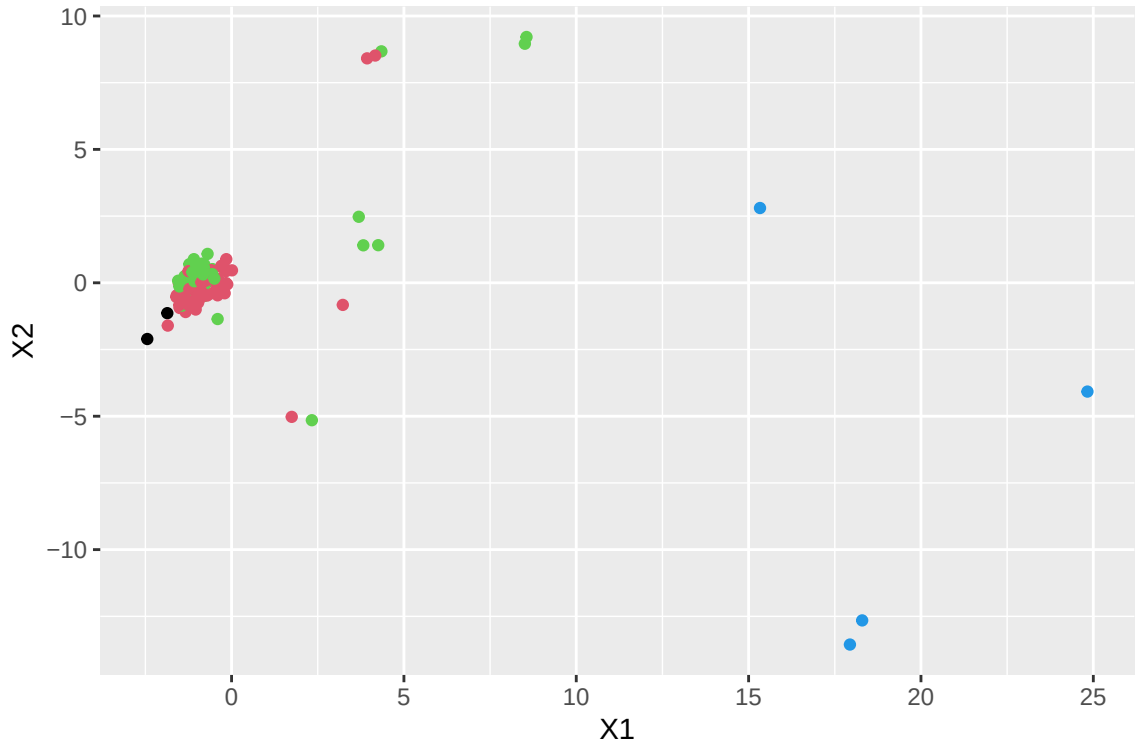
3.5 Wnioski do zadania 2

4 Zadanie 3

Wybranym przez nas zbiorem danych jest ...

Wczytajmy dane i przygotujmy je do do skalowania wielowymiarowego.

```
data <- read.csv("mds_data.csv")
data.for.mds <- as.matrix(daisy(data[-data$class], stand=T))
mds.k <- isoMDS(data.for.mds, k = 2, trace = FALSE)
after.sammon <- mds.k$points
ggplot(data = data.frame(after.sammon), aes(x= X1, y=X2)) + geom_point(color = data$clas
```

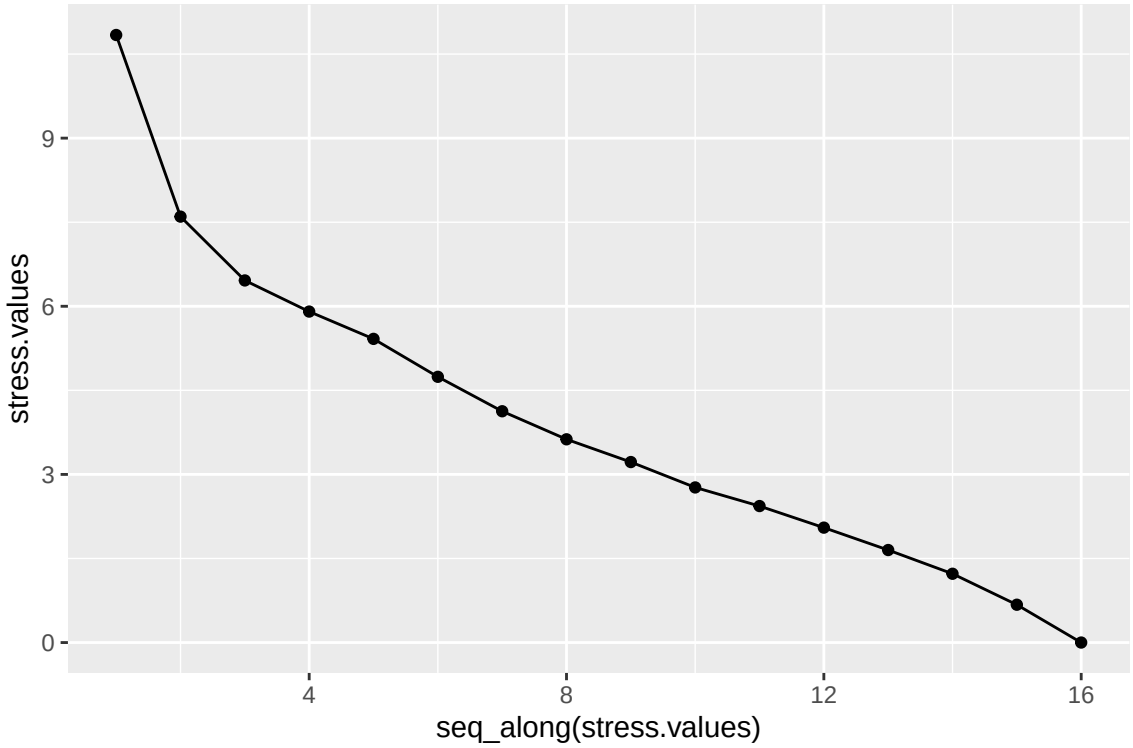


Porównamy teraz jakoś odwzorowania MDS w zależności od wielkości wymiaru d przestrzeni docelowej. Przedstawimy na wykresie wartości funkcji STRESS, jak i wykonamy diagramy Sheparda.

```
d.max <- 16
stress.values <- numeric(d.max)

for (d in 1:d.max) {
  mds.k <- isoMDS(data.for.mds, k = d, trace = FALSE)
  stress.values[d] <- mds.k$stress;
}

stress.values <- data.frame(stress.values)
ggplot(data = stress.values, aes(x=seq_along(stress.values), y=stress.values)) + geom_po
```


stress.values
9
6
3
0
4
8
12
16
seq_along(stress.values)