Raport 2

Eksploracja danych

Mikołaj Langner, Marcin Kostrzewa
nr albumów: 255716, 255749

2021-04-19

Spis treści

1 Wstęp **1**

2 Zadanie 1 **1**

3 Zadanie 2 **21**
3.1 Wczytanie i przygotowanie danych 21
3.2 Składowe główne i ich analiza 21
3.3 Wizualizacja danych 24
3.4 Korelacja zmiennych 25
3.5 Wnioski do zadania 2 25

4 Zadanie 3 **25**

1 Wstęp

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

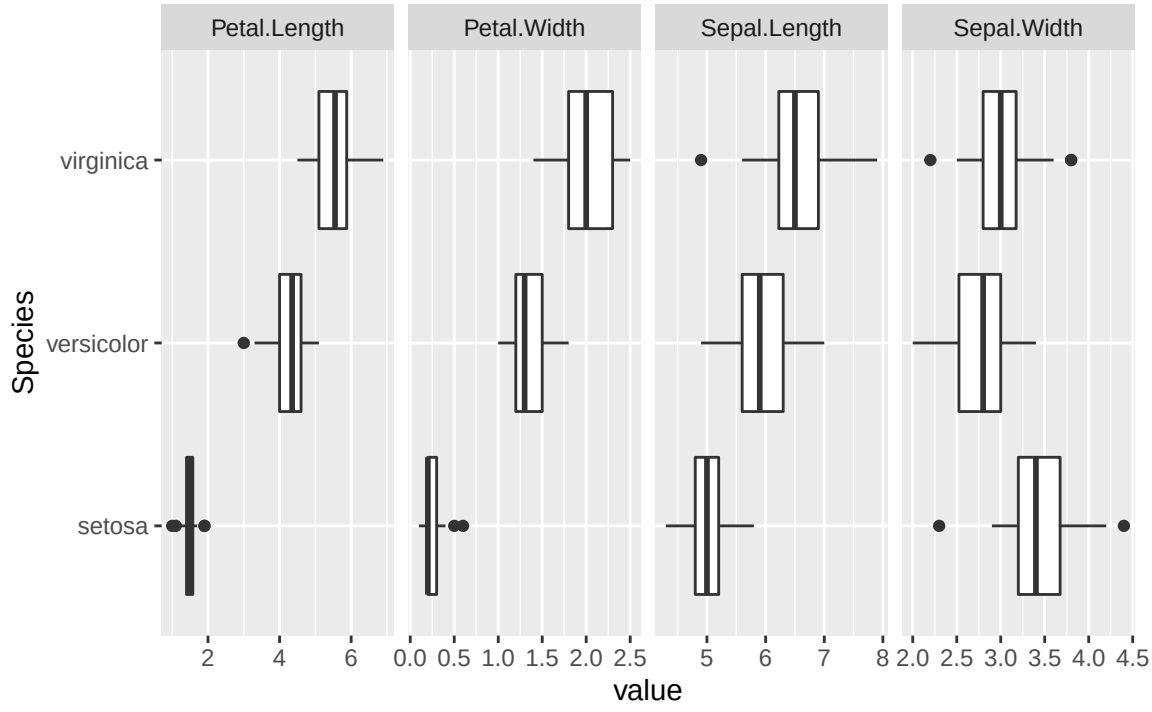
2 Zadanie 1

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

```
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")
```




```
aggregate(. ~ Species, iris, cv)
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa   0.07041344   0.1105789   0.11878522   0.4283967
## 2 versicolor   0.08695606   0.1132846   0.11030774   0.1491348
## 3  virginica   0.09652089   0.1084387   0.09940466   0.1355627
```

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

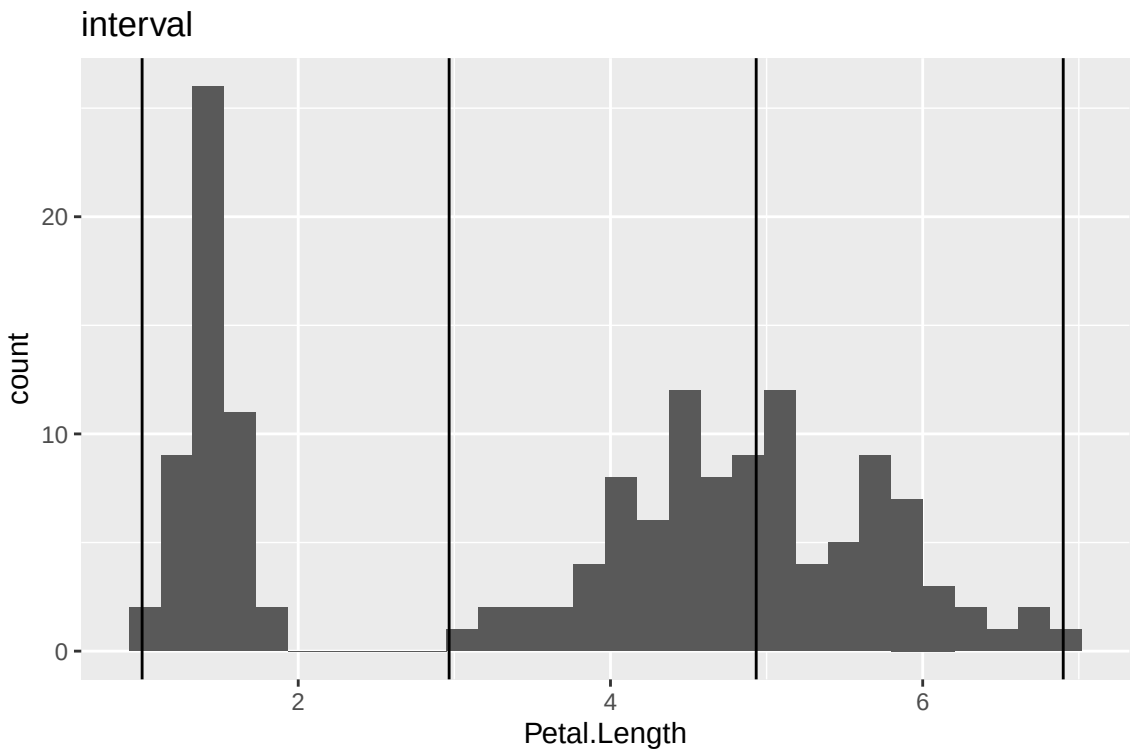
Porównamy ze sobą cztery metody dyskretyzacji nienadzorowanej:

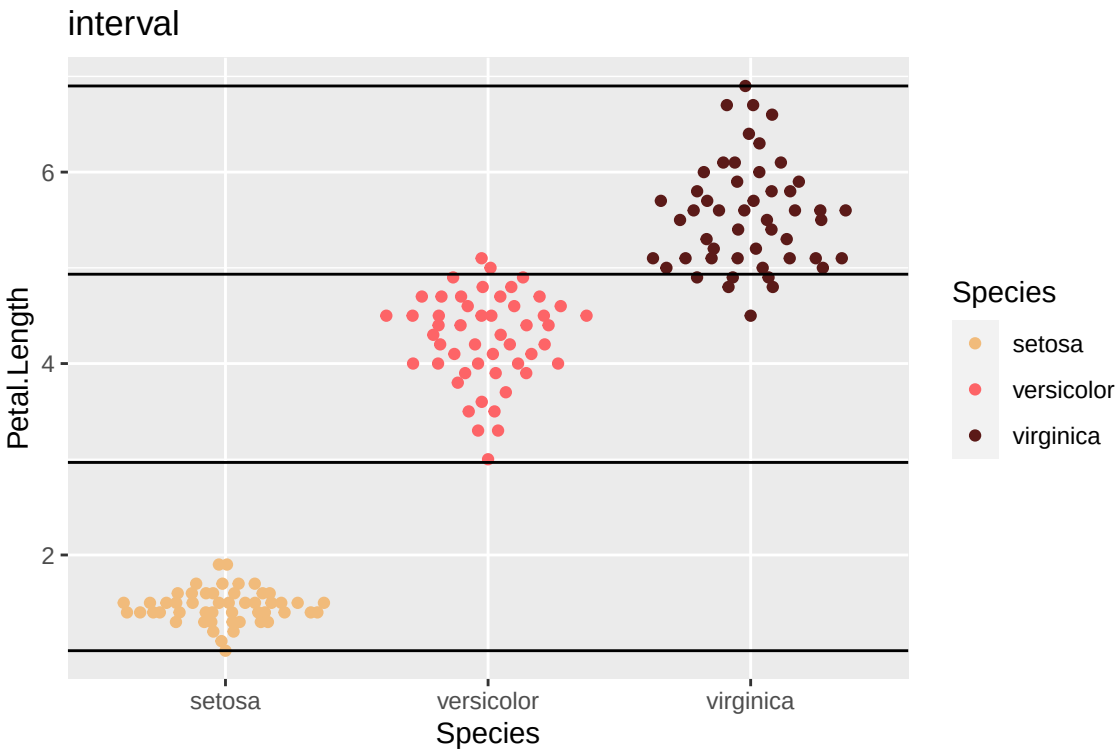
- equal width,
- equal frequency,
- k-means clustering,
- dyskretyzację dla przedziałów zadanych przez użytkownika.

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

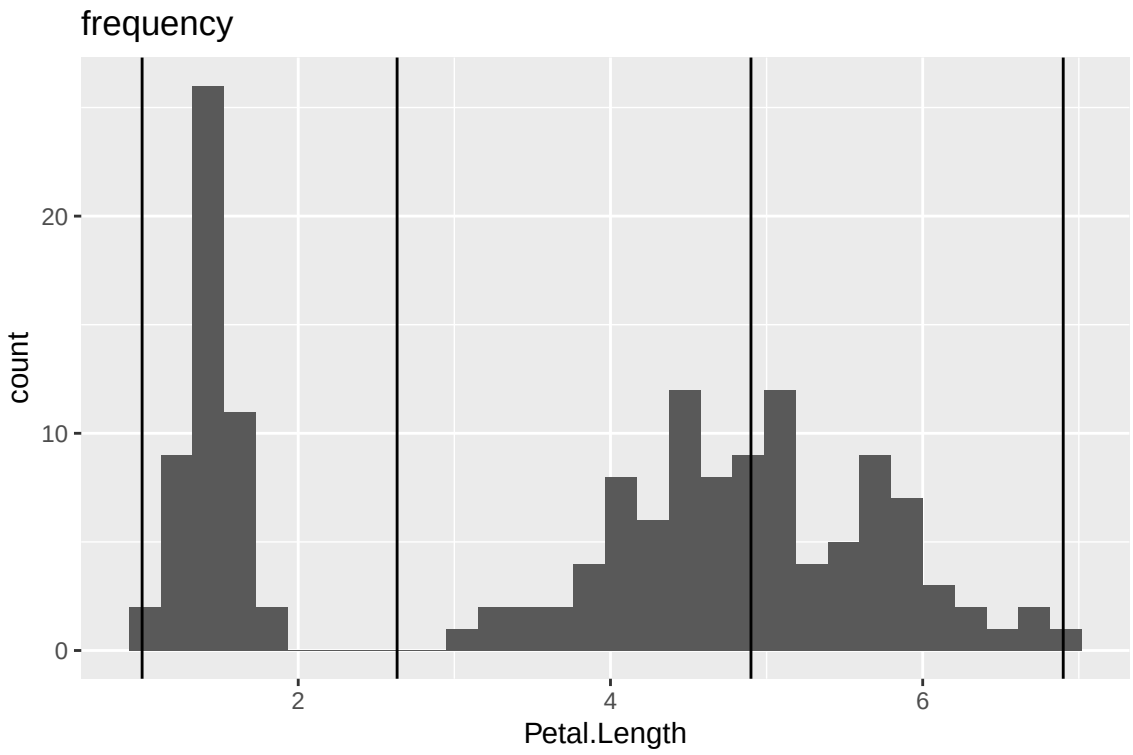
```
intervals <- c(min(iris$Petal.Length), 2, 5, max(iris$Petal.Length))
for (method in c("interval", "frequency", "cluster", "fixed")) {
  petal.length.discretized <- if (method != "fixed")
    discretize(iris$Petal.Length, method=method) else
    discretize(iris$Petal.Length, method=method, breaks=intervals)
```

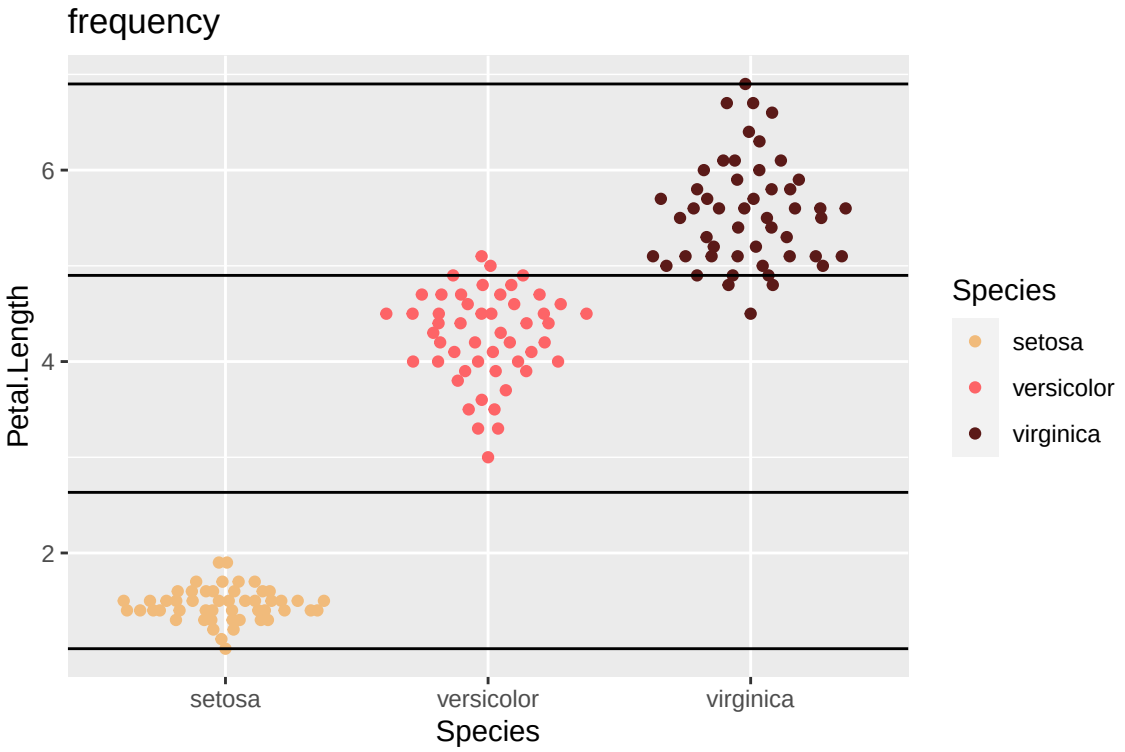
```
  print(ggplot(iris, aes(Petal.Length)) +
          geom_histogram() +
          geom_vline(xintercept=attributes(petal.length.discretized)$"discretized:breaks
          ggtitle(method))
  print(ggplot(iris, aes(Species, Petal.Length)) +
    geom_quasirandom(aes(col=Species)) +
    scale_color_manual(values=wes_palette("GrandBudapest1", 3)) +
    geom_hline(yintercept=attributes(petal.length.discretized)$"discretized:breaks") +
    ggtitle(method))
  discretized.table <- table(petal.length.discretized, iris$Species)
  matchClasses(discretized.table)
}
```



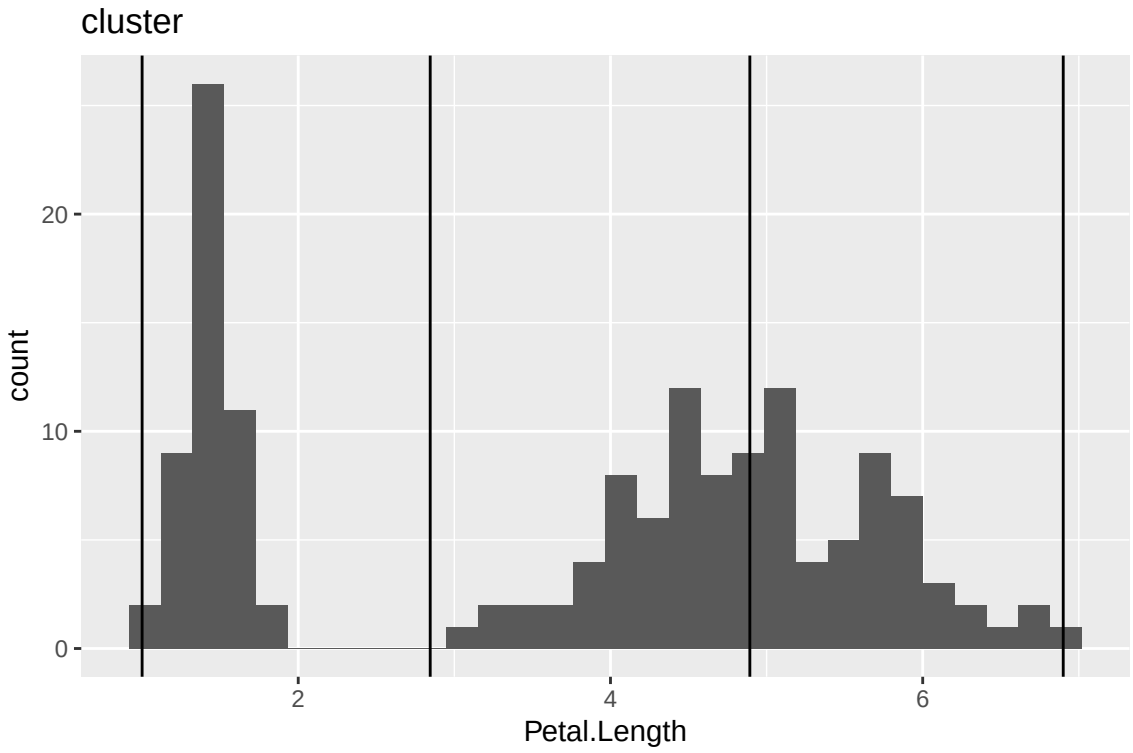


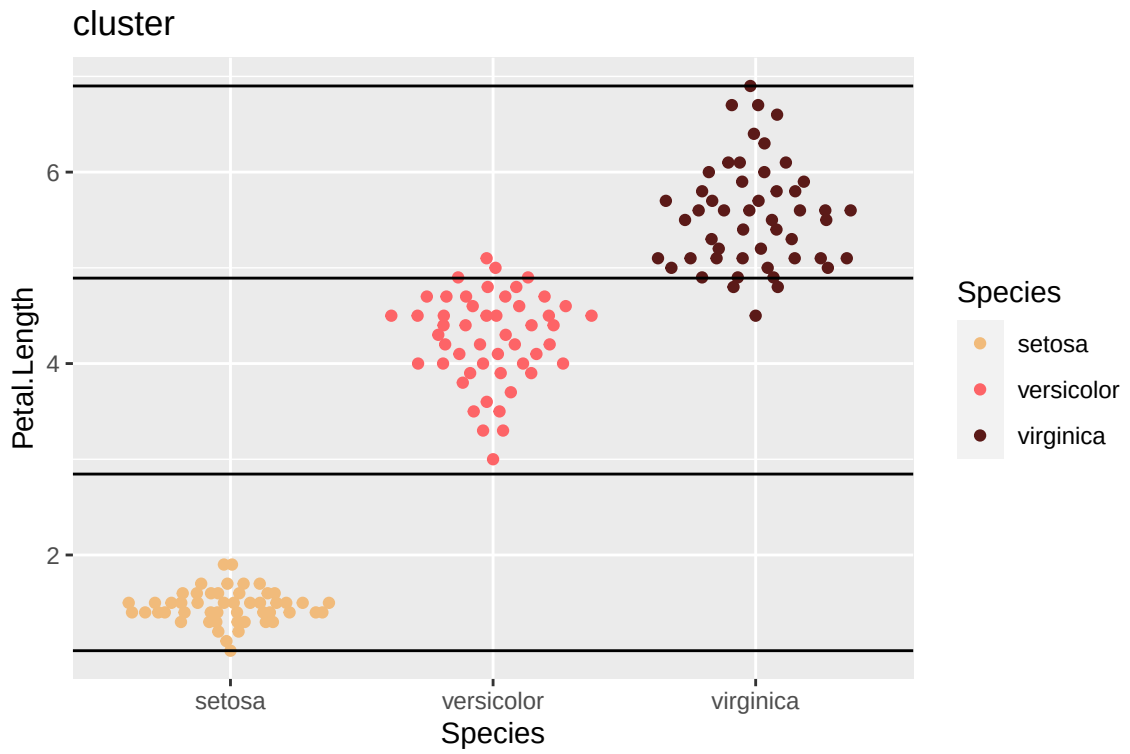

```
## Cases in matched pairs: 94.67 %
```



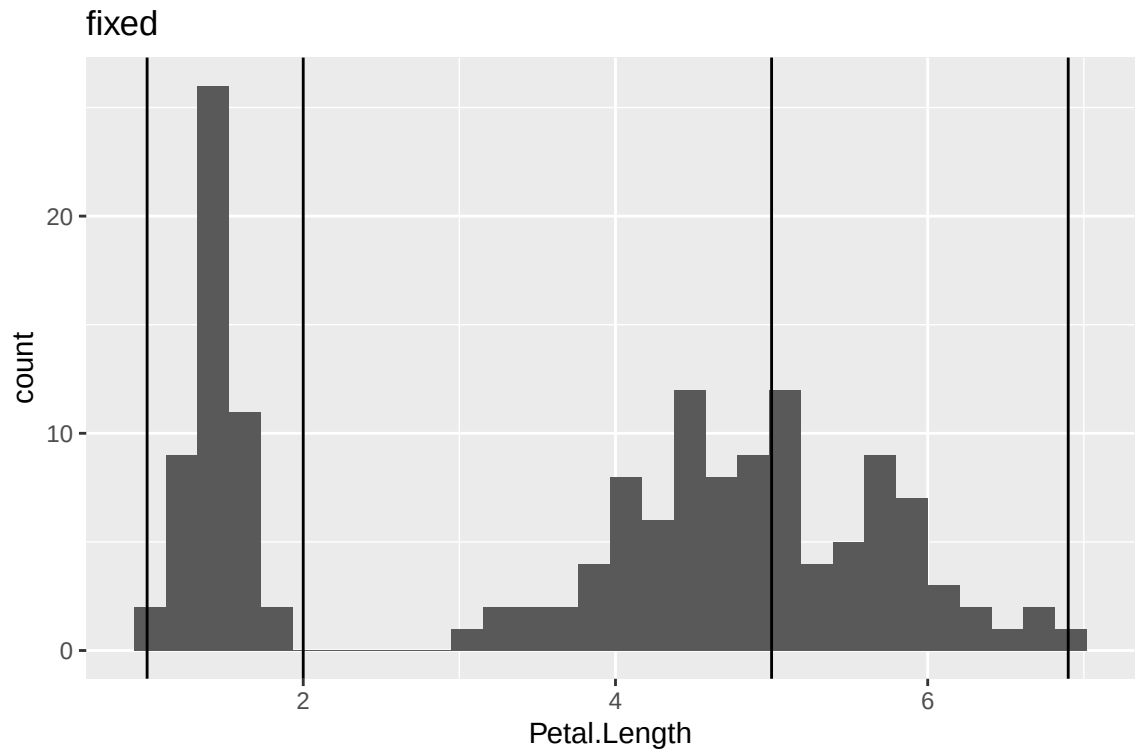


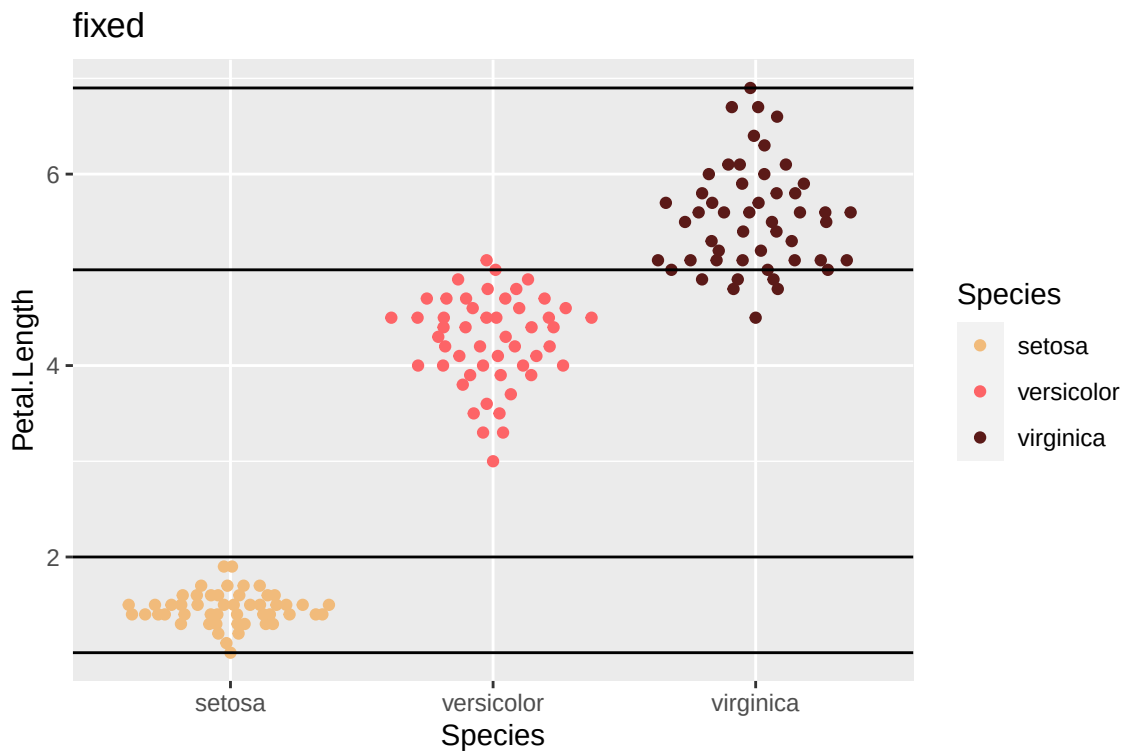

```
## Cases in matched pairs: 95.33 %
```






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

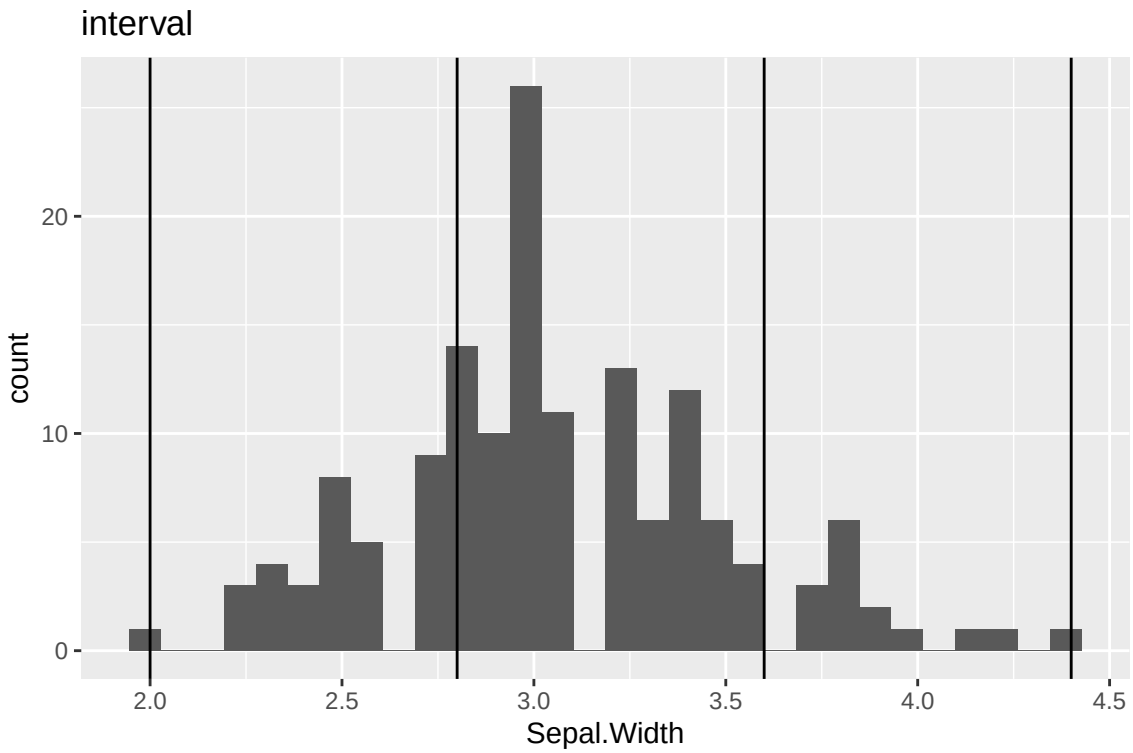


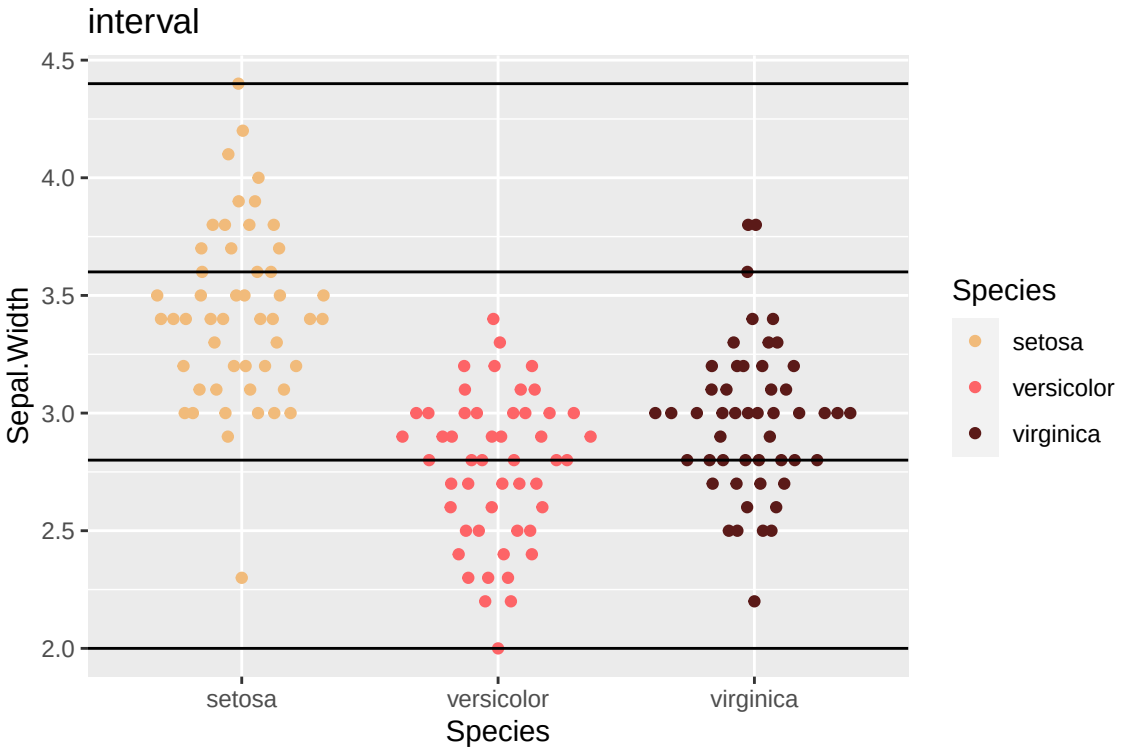



```
## Cases in matched pairs: 94.67 %
```

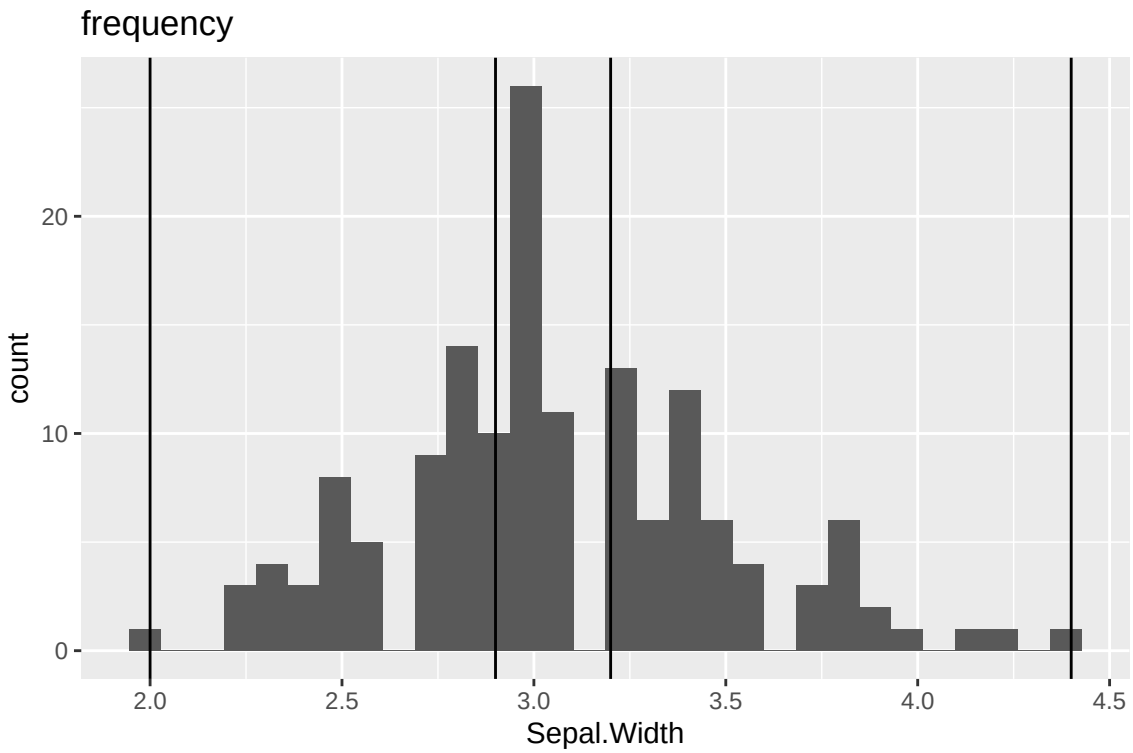
Możemy zobaczyć teraz jak poszczególne metody działają dla zmiennej Sepal.Width, która najgorzej radzi sobie z rozdzielaniem gatunków.

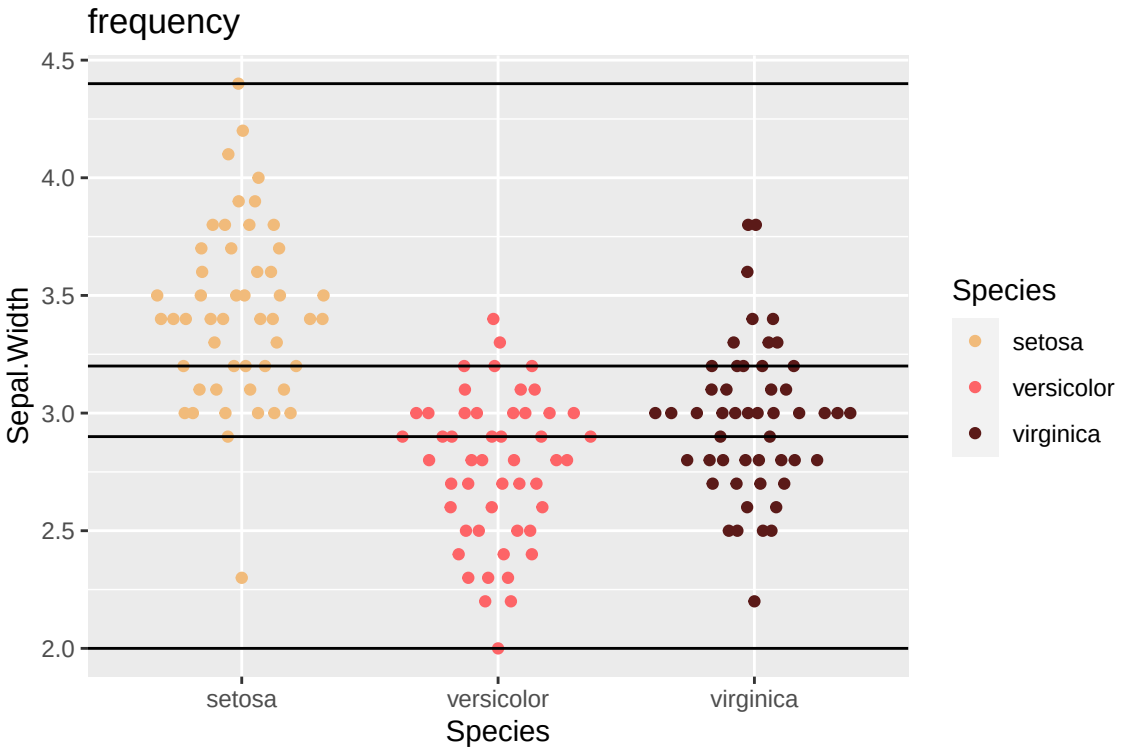
```
intervals <- c(min(iris$Sepal.Width), 2.5, 3, max(iris$Sepal.Width))
for (method in c("interval", "frequency", "cluster", "fixed")) {
  sepal.width.discretized <- if (method != "fixed")
    discretize(iris$Sepal.Width, method=method) else
    discretize(iris$Sepal.Width, method=method, breaks=intervals)
  print(ggplot(iris, aes(Sepal.Width)) +
          geom_histogram() +
          geom_vline(xintercept=attributes(sepal.width.discretized)$"discretized:breaks"
          ggtitle(method))
  print(ggplot(iris, aes(Species, Sepal.Width)) +
    geom_quasirandom(aes(col=Species)) +
    scale_color_manual(values=wes_palette("GrandBudapest1", 3)) +
    geom_hline(yintercept=attributes(sepal.width.discretized)$"discretized:breaks") +
    ggtitle(method))
  discretized.table <- table(sepal.width.discretized, iris$Species)
  matchClasses(discretized.table)
}
```



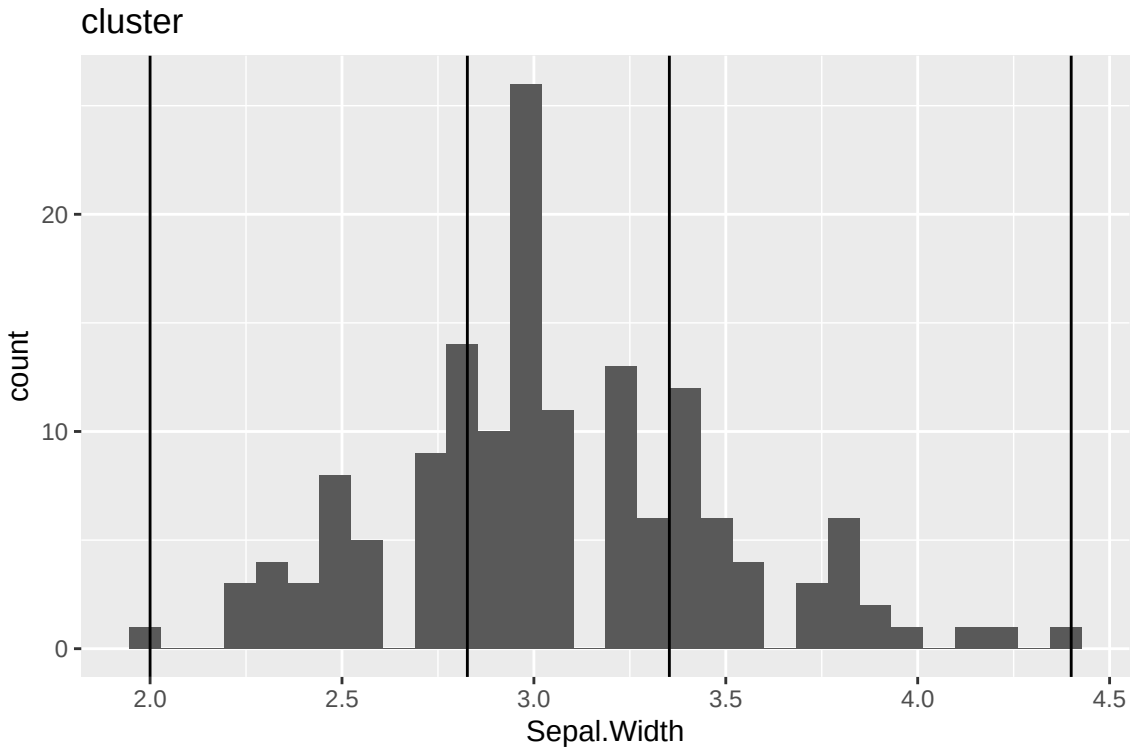


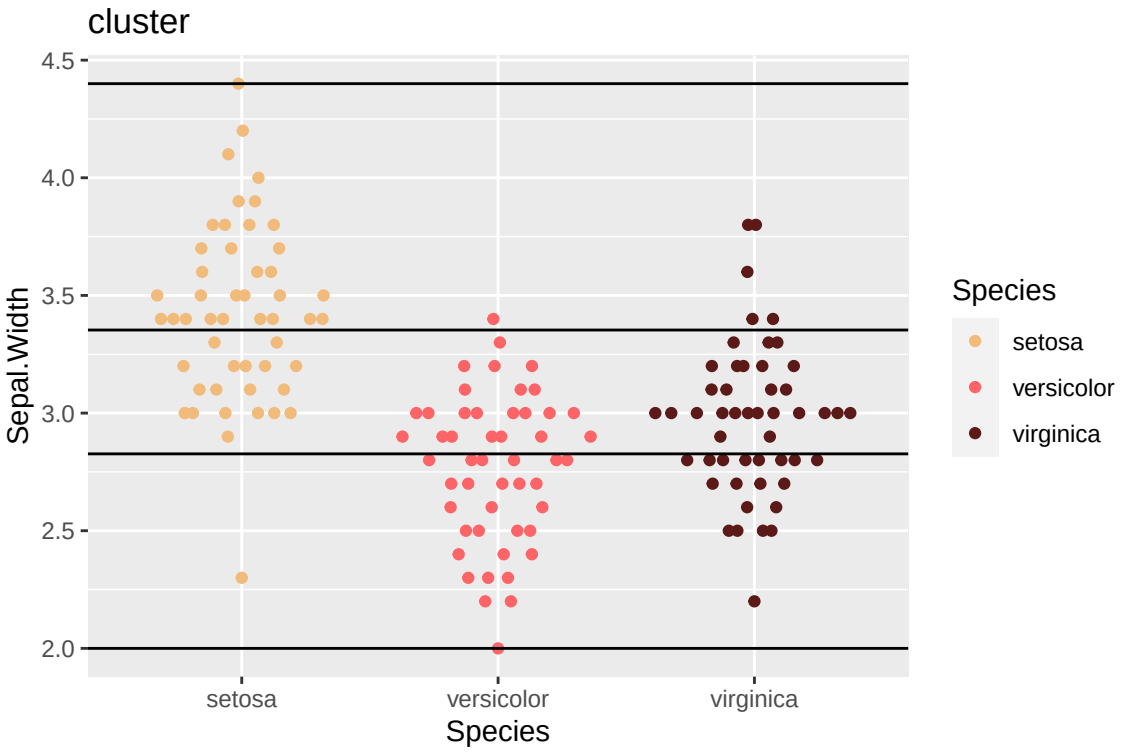

```
## Cases in matched pairs: 50.67 %
```



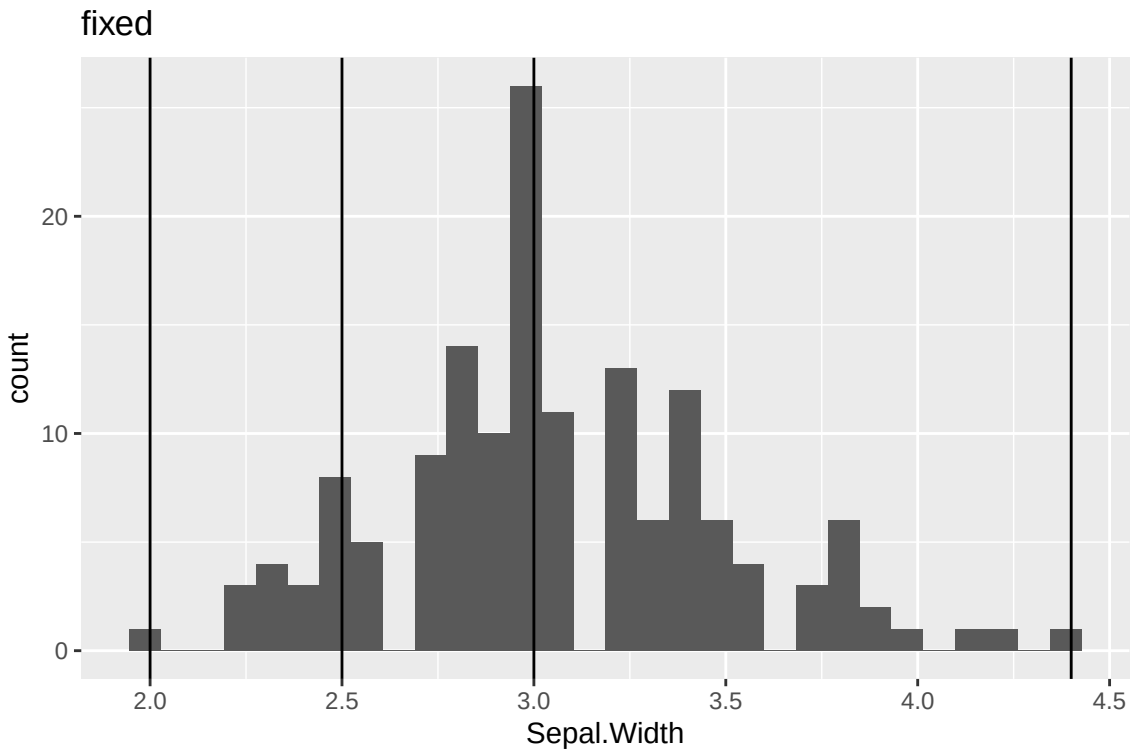


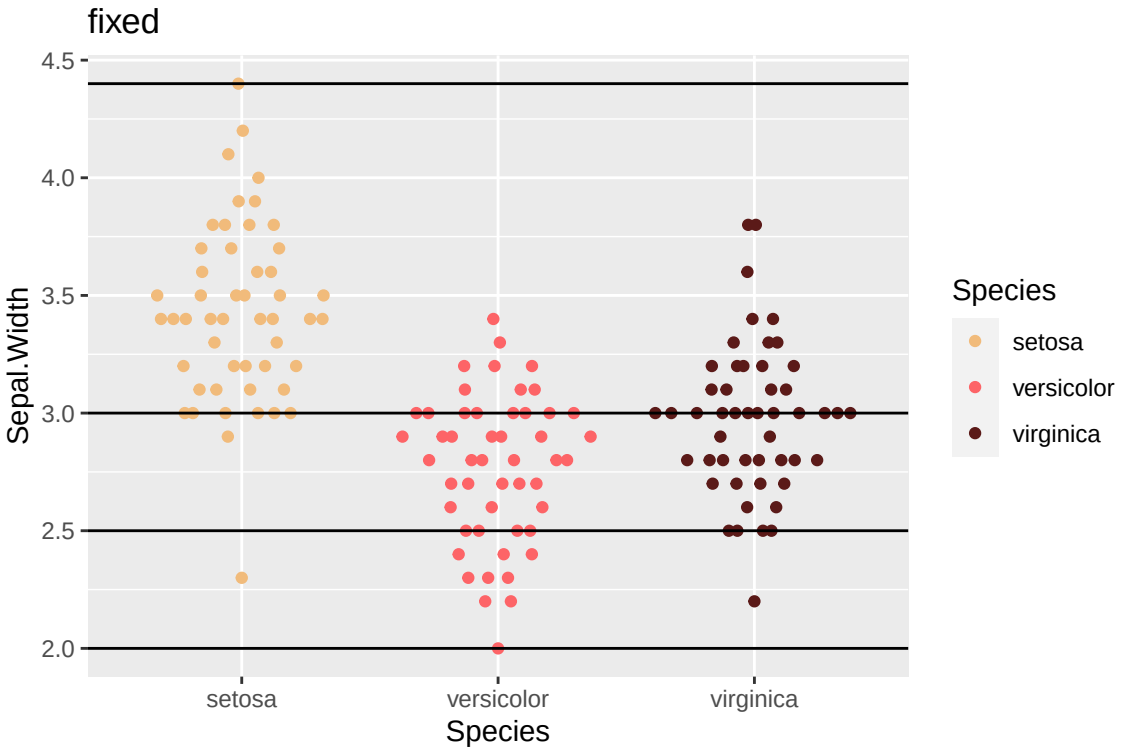

```
## Cases in matched pairs: 55.33 %
```






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

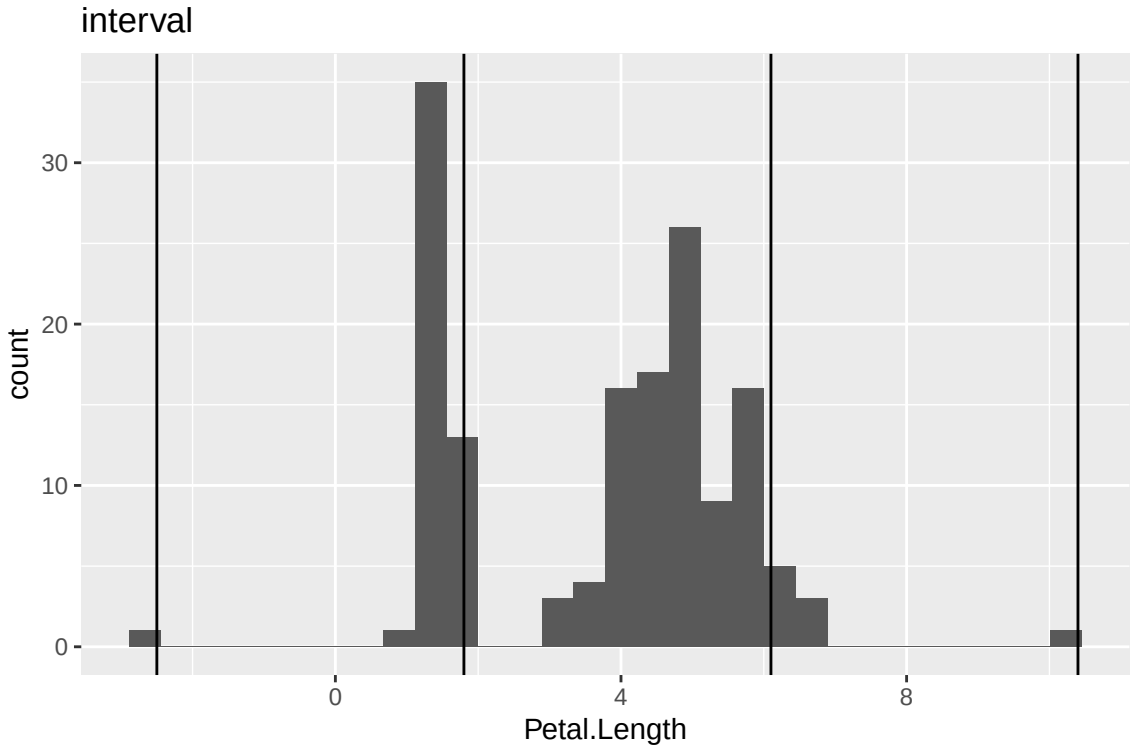


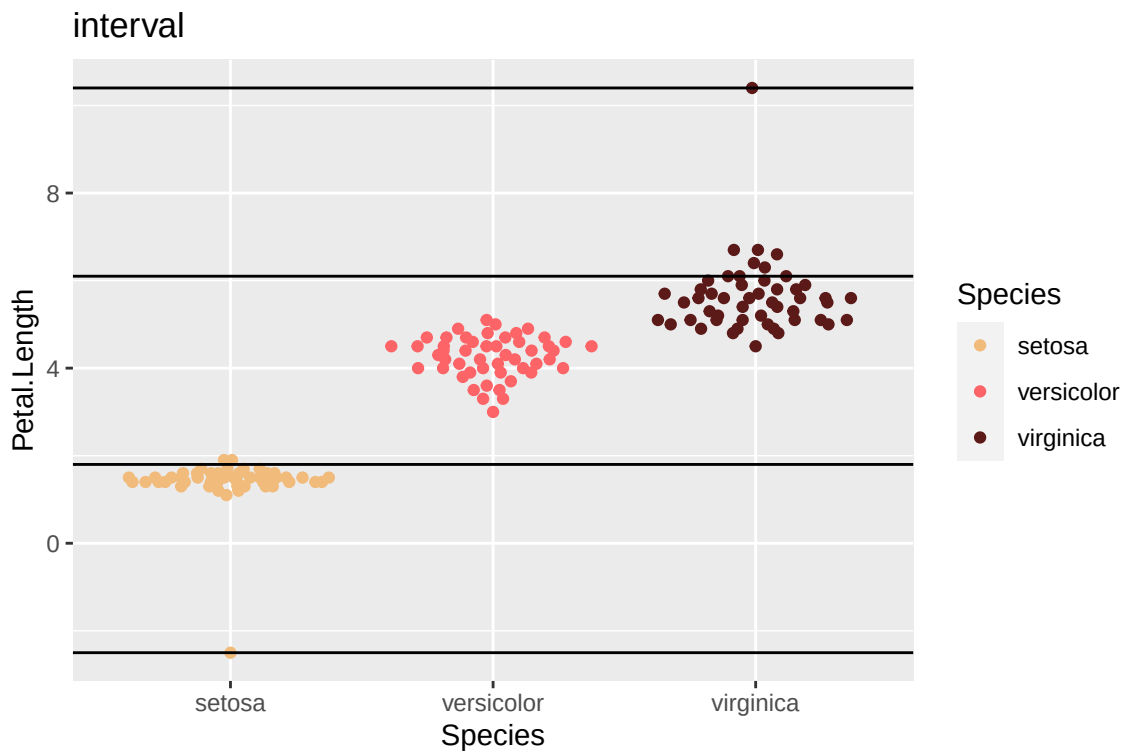



```
## Cases in matched pairs: 54.67 %
```

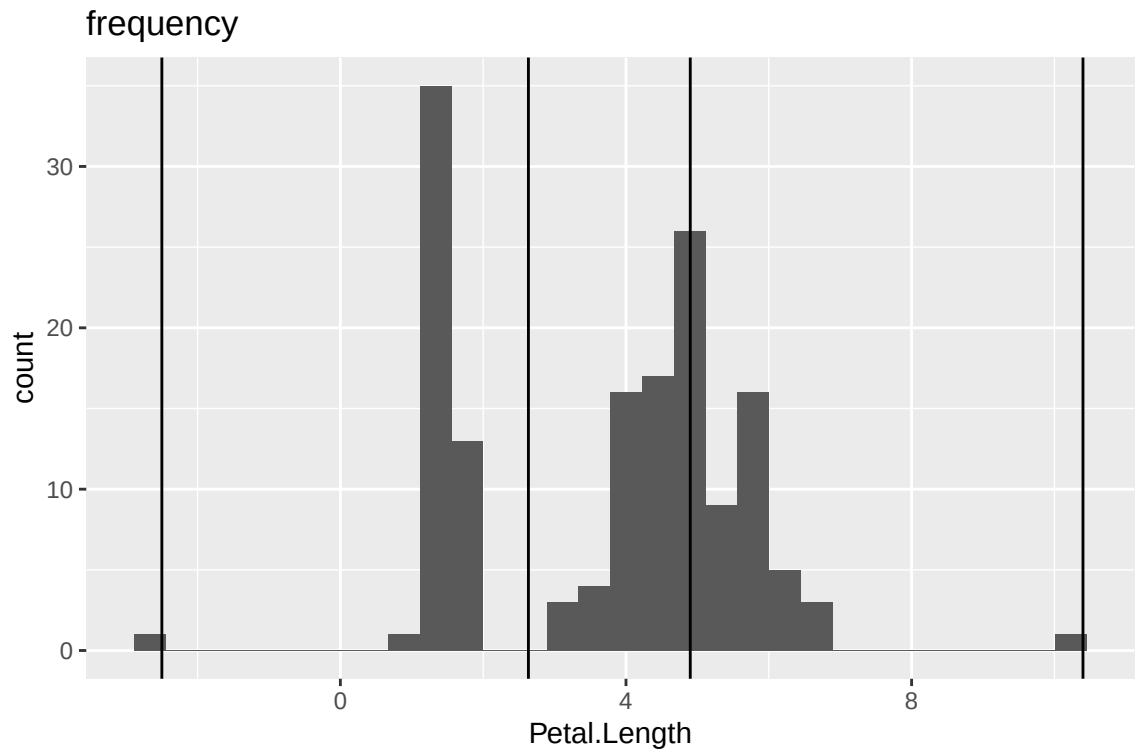
Rozpatrzmy znowu dyskretyzację obu zmiennych przy dodaniu sztucznie wartości odstających. Przeprowadźmy najpierw dyskretyzację dla zmiennej Petal.Length.

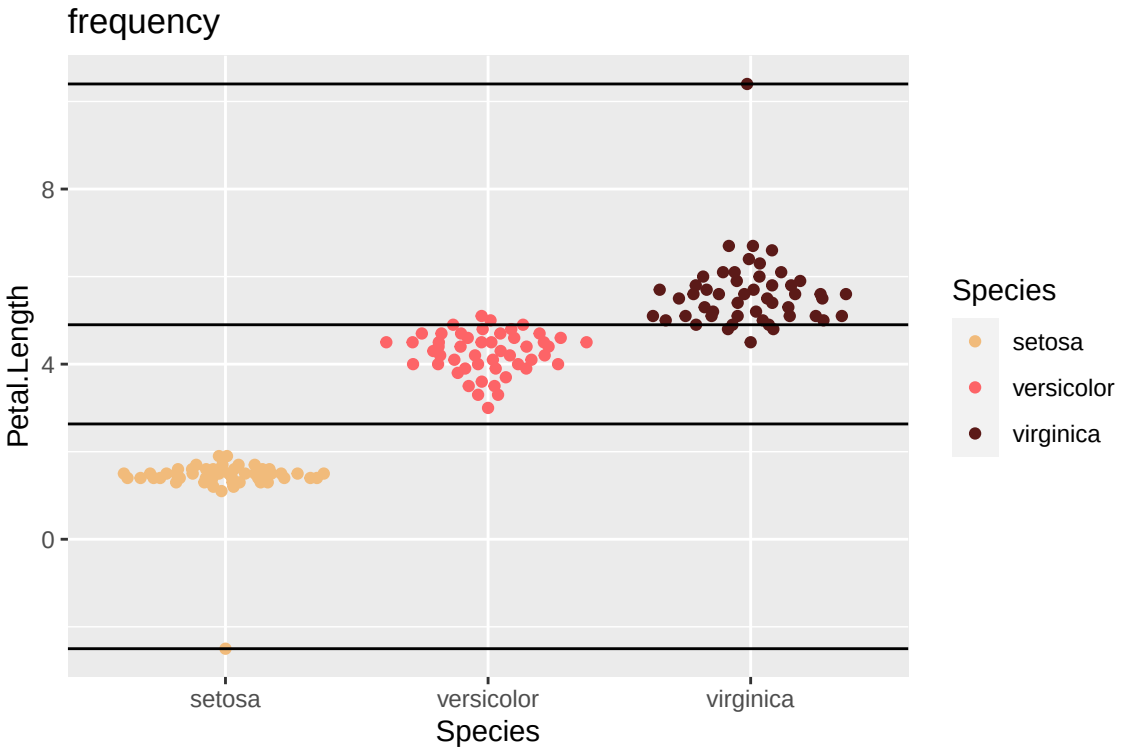
```
iris$Petal.Length[which.min(iris$Petal.Length)] <- min(iris$Petal.Length) - IQR(iris$Pet
iris$Petal.Length[which.max(iris$Petal.Length)] <- max(iris$Petal.Length) + IQR(iris$Pet
intervals <- c(min(iris$Petal.Length), 2, 5, max(iris$Petal.Length))
for (method in c("interval", "frequency", "cluster", "fixed")) {
  petal.length.discretized <- if (method != "fixed")
    discretize(iris$Petal.Length, method=method) else
    discretize(iris$Petal.Length, method=method, breaks=intervals)
  print(ggplot(iris, aes(Petal.Length)) +
          geom_histogram() +
          geom_vline(xintercept=attributes(petal.length.discretized)$"discretized:breaks
          ggtitle(method))
  print(ggplot(iris, aes(Species, Petal.Length)) +
    geom_quasirandom(aes(col=Species)) +
    scale_color_manual(values=wes_palette("GrandBudapest1", 3)) +
    geom_hline(yintercept=attributes(petal.length.discretized)$"discretized:breaks") +
    ggtitle(method))
  discretized.table <- table(petal.length.discretized, iris$Species)
  matchClasses(discretized.table)
}
```



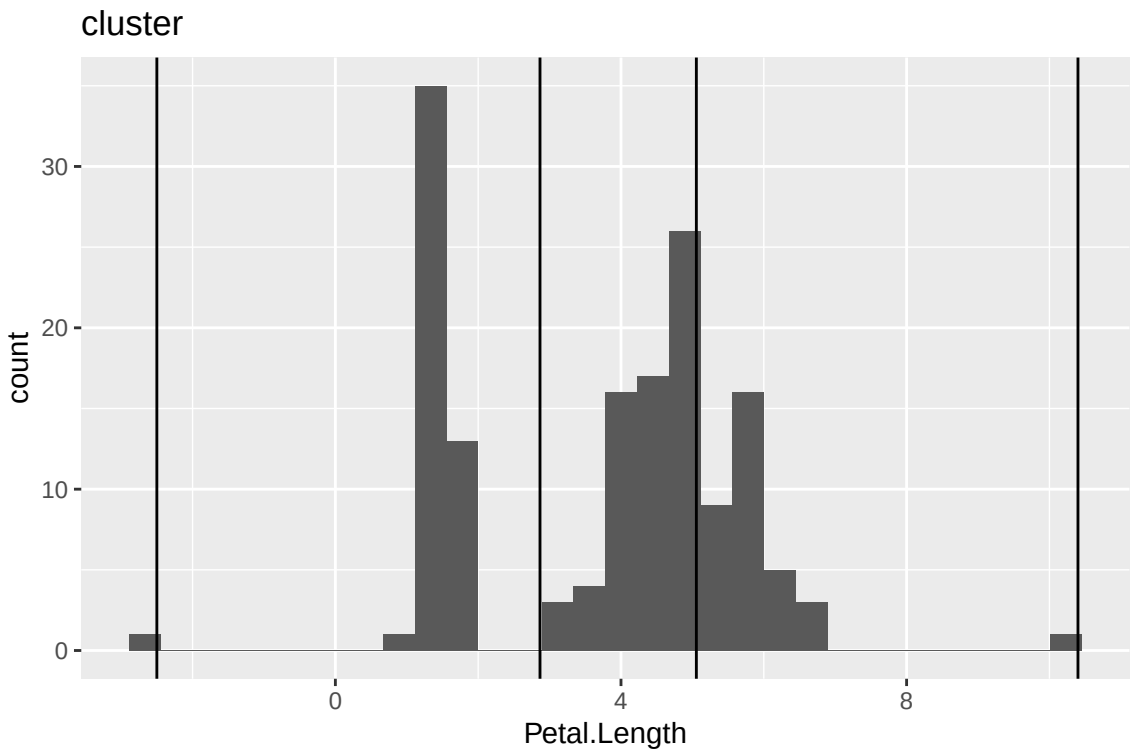


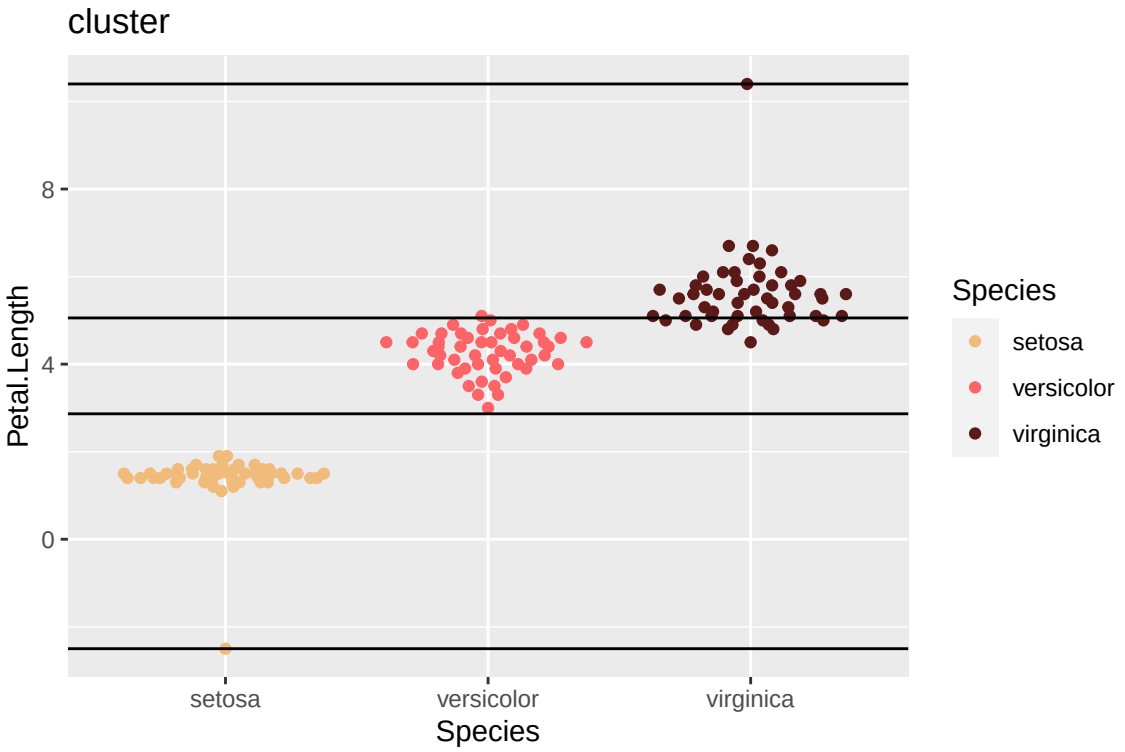

```
## Cases in matched pairs: 71.33 %
```



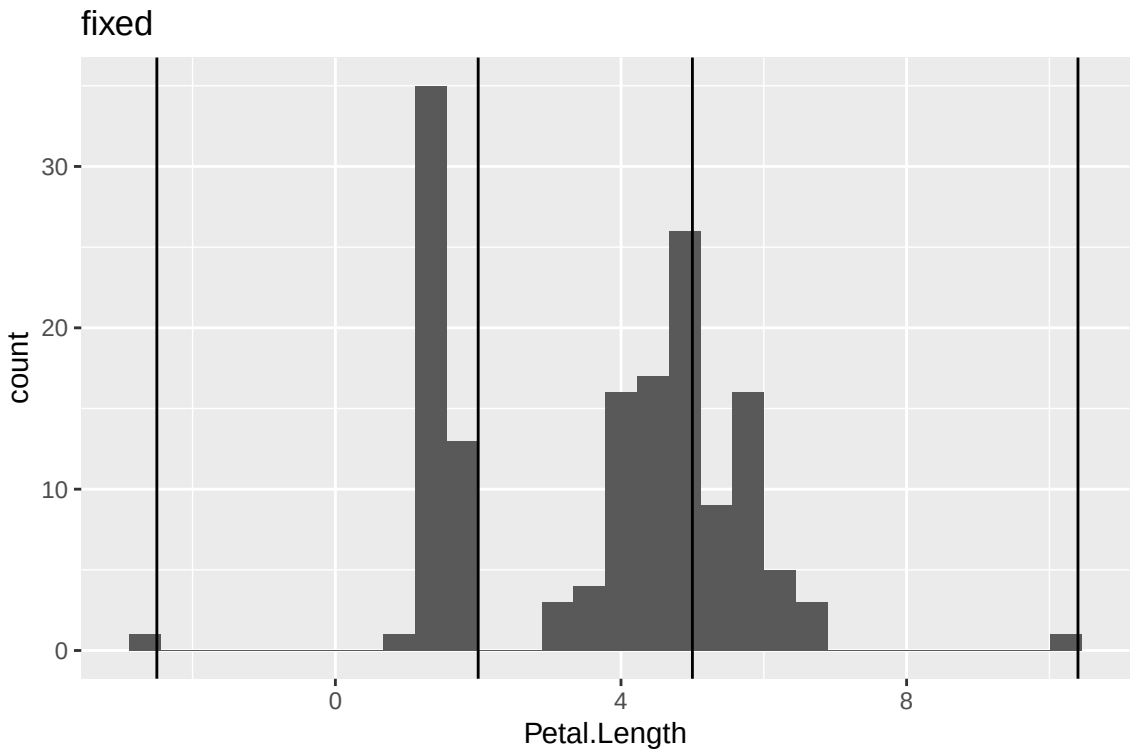


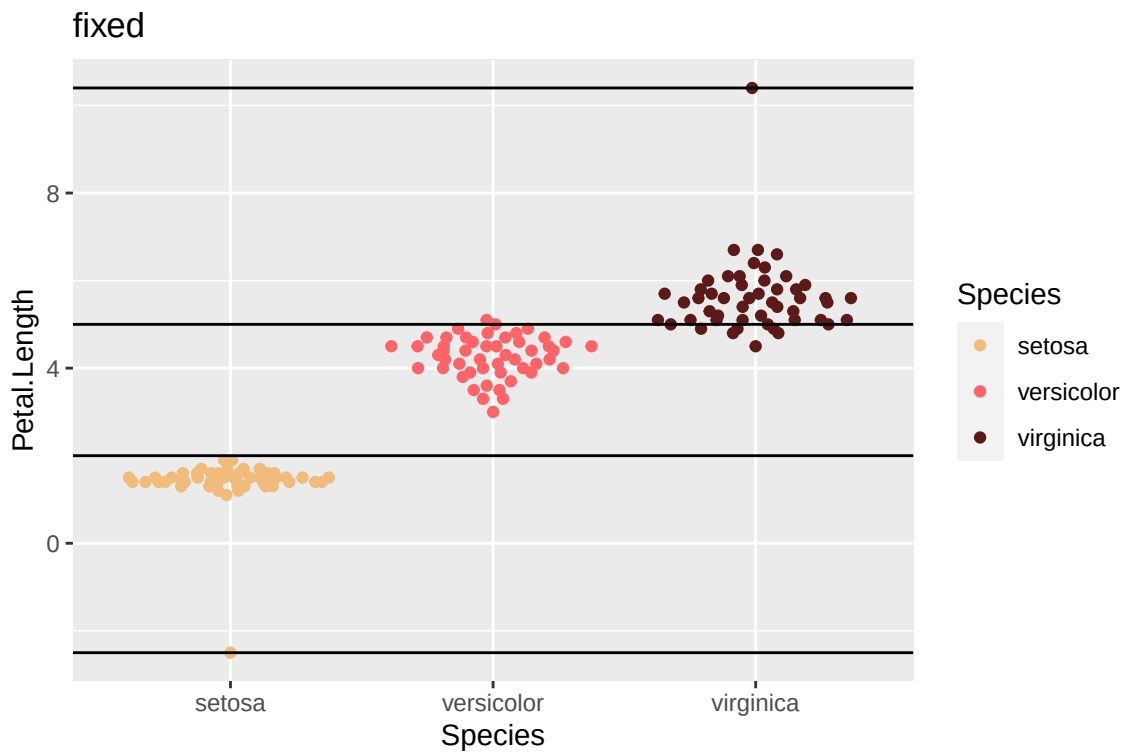

```
## Cases in matched pairs: 95.33 %
```






```
## Cases in matched pairs: 93.33 %
```

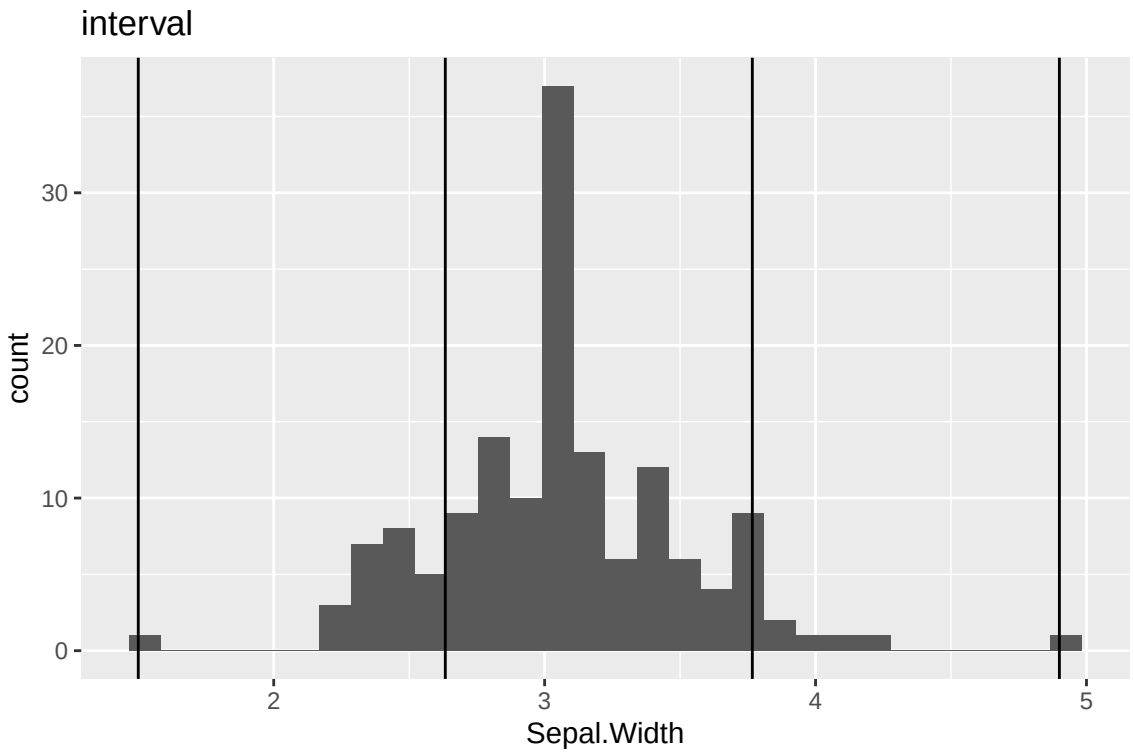


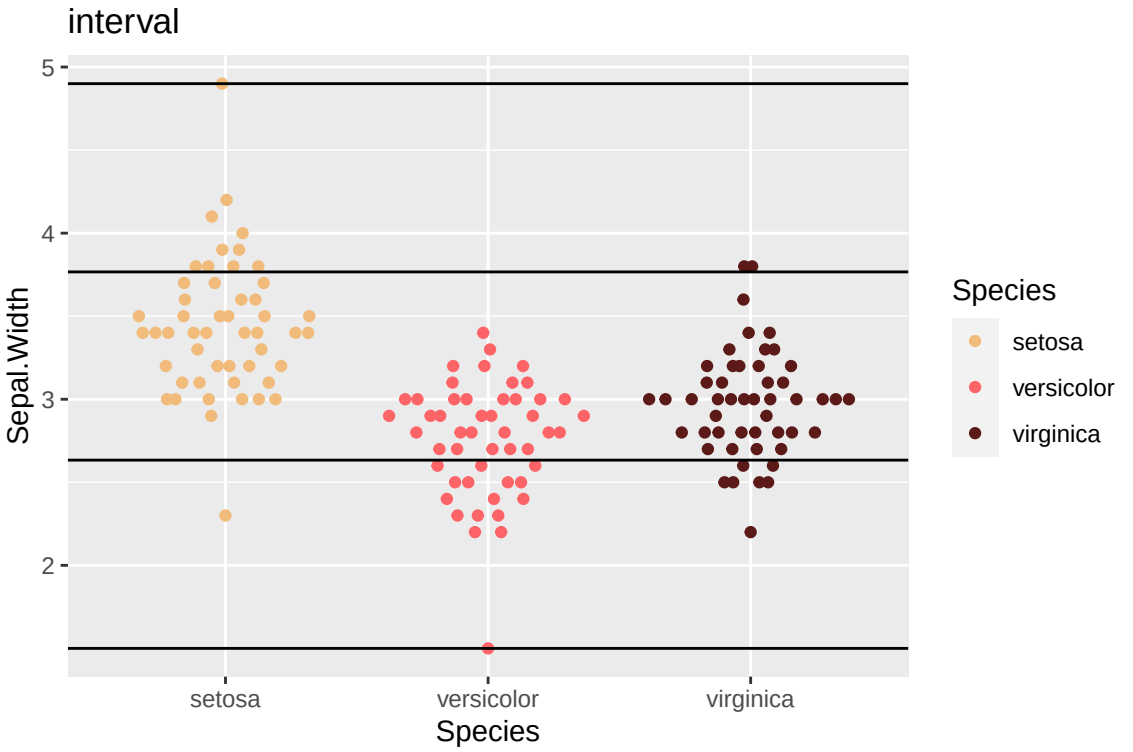



```
## Cases in matched pairs: 94.67 %
```

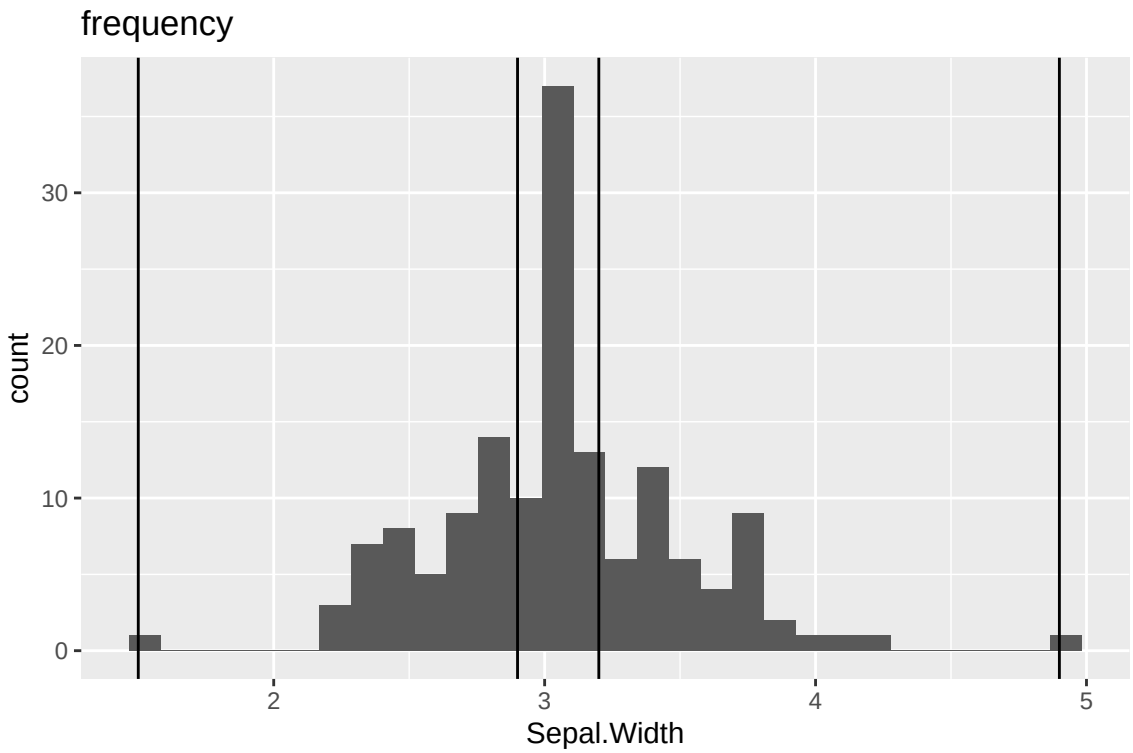
Dla zmiennej Sepal.Width po dodaniu wartości odstających dyskretyzacja wygląda następująco:

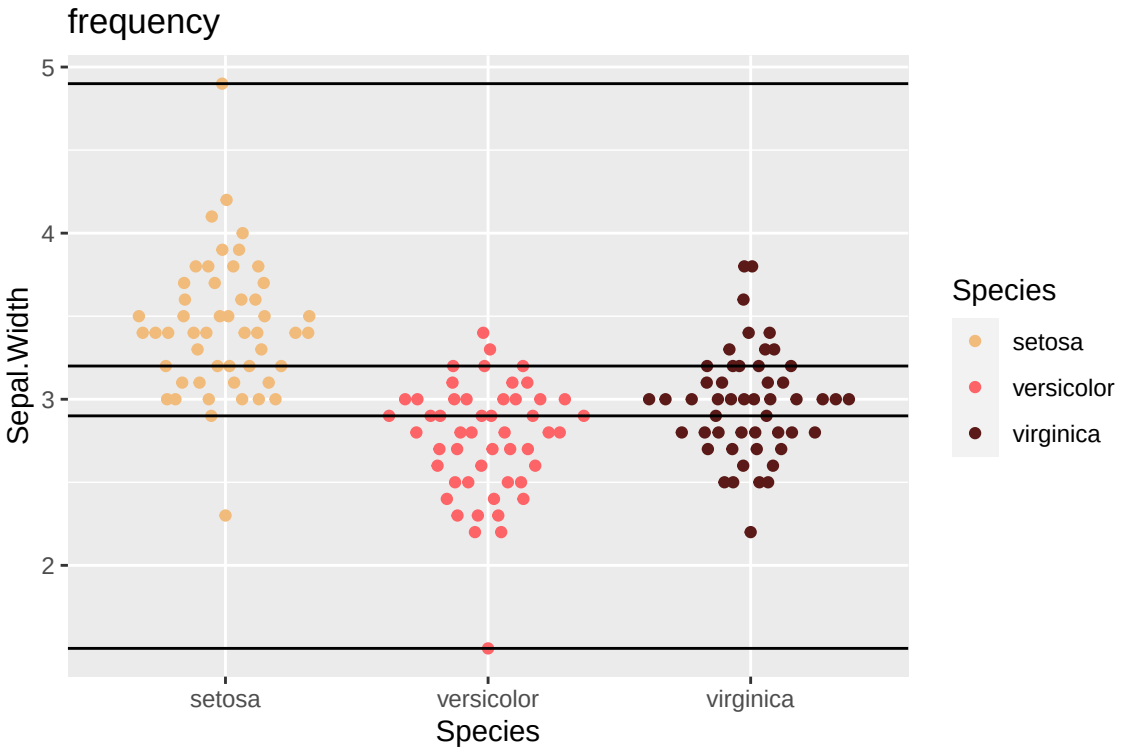
```
iris$Sepal.Width[which.min(iris$Sepal.Width)] <- min(iris$Sepal.Width) - IQR(iris$Sepal
iris$Sepal.Width[which.max(iris$Sepal.Width)] <- max(iris$Sepal.Width) + IQR(iris$Sepal
intervals <- c(min(iris$Sepal.Width), 2.5, 3, max(iris$Sepal.Width))
for (method in c("interval", "frequency", "cluster", "fixed")) {
  sepal.width.discretized <- if (method != "fixed")
    discretize(iris$Sepal.Width, method=method) else
    discretize(iris$Sepal.Width, method=method, breaks=intervals)
  print(ggplot(iris, aes(Sepal.Width)) +
          geom_histogram() +
          geom_vline(xintercept=attributes(sepal.width.discretized)$"discretized:breaks"
          ggtitle(method))
  print(ggplot(iris, aes(Species, Sepal.Width)) +
    geom_quasirandom(aes(col=Species)) +
    scale_color_manual(values=wes_palette("GrandBudapest1", 3)) +
    geom_hline(yintercept=attributes(sepal.width.discretized)$"discretized:breaks") +
    ggtitle(method))
  discretized.table <- table(sepal.width.discretized, iris$Species)
  matchClasses(discretized.table)
}
```



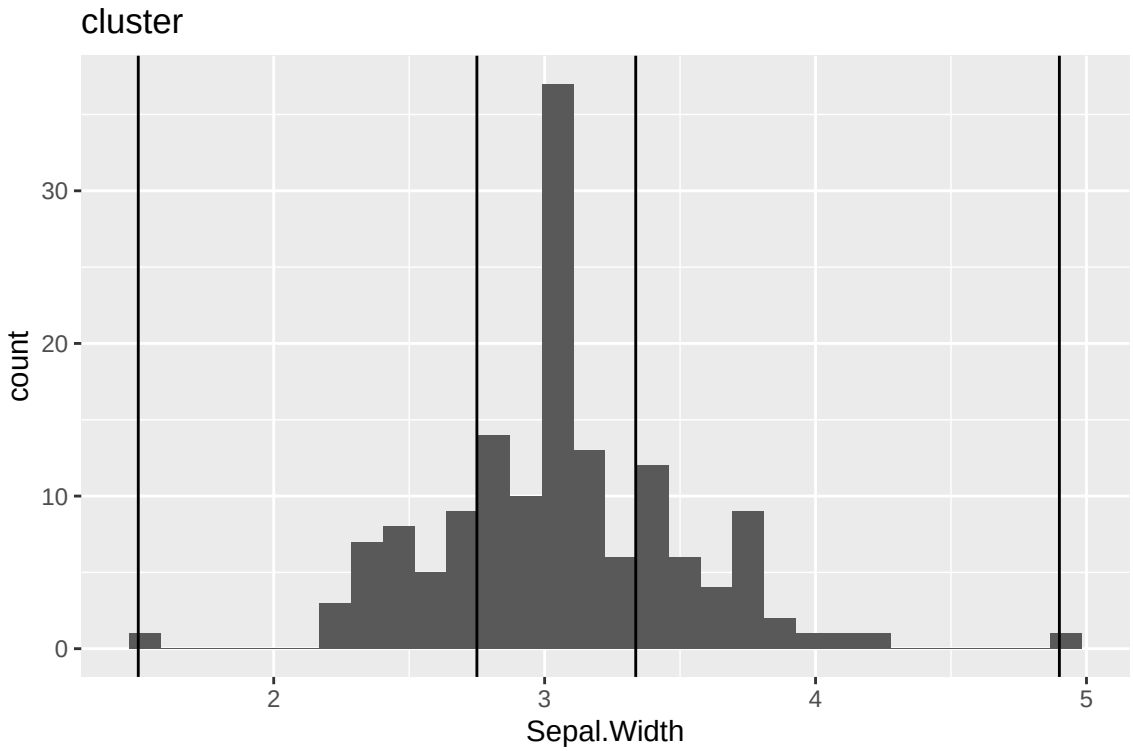


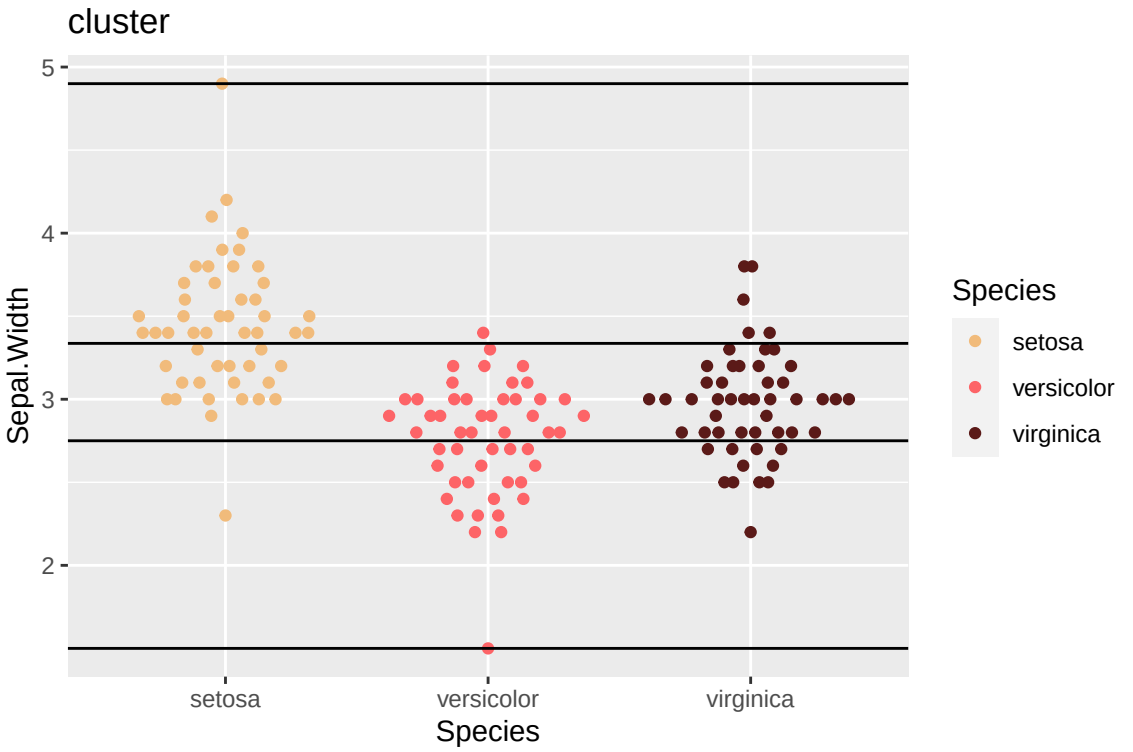

```
## Cases in matched pairs: 44.67 %
```



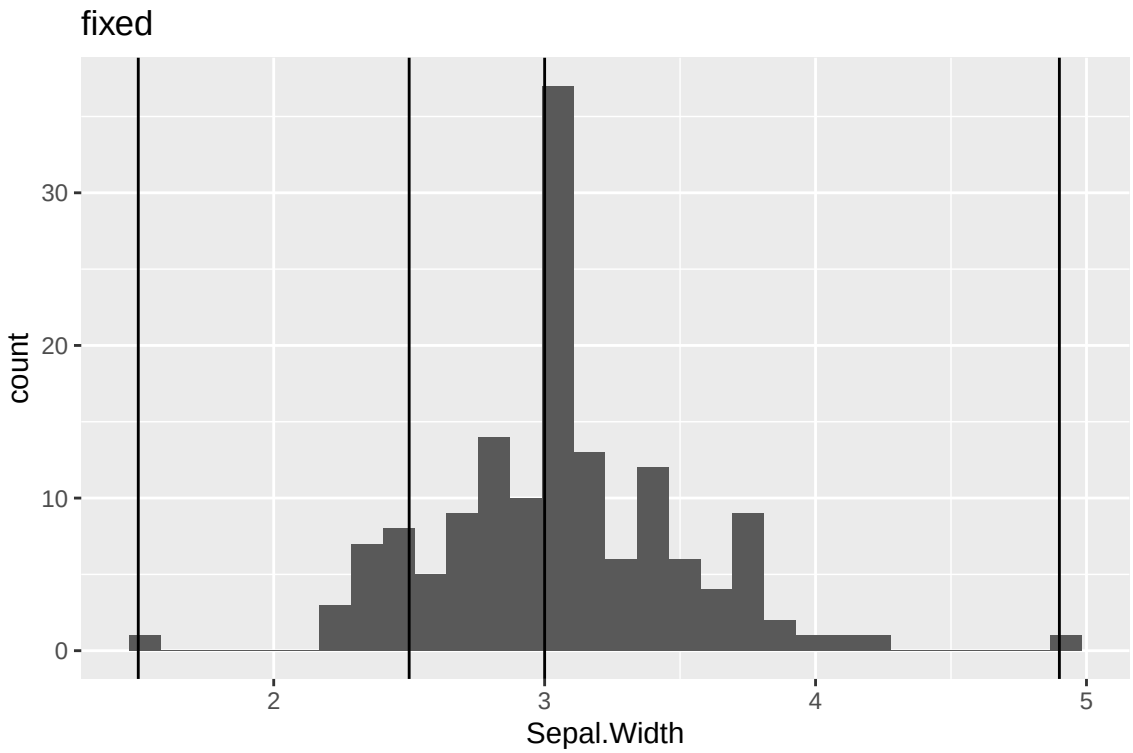


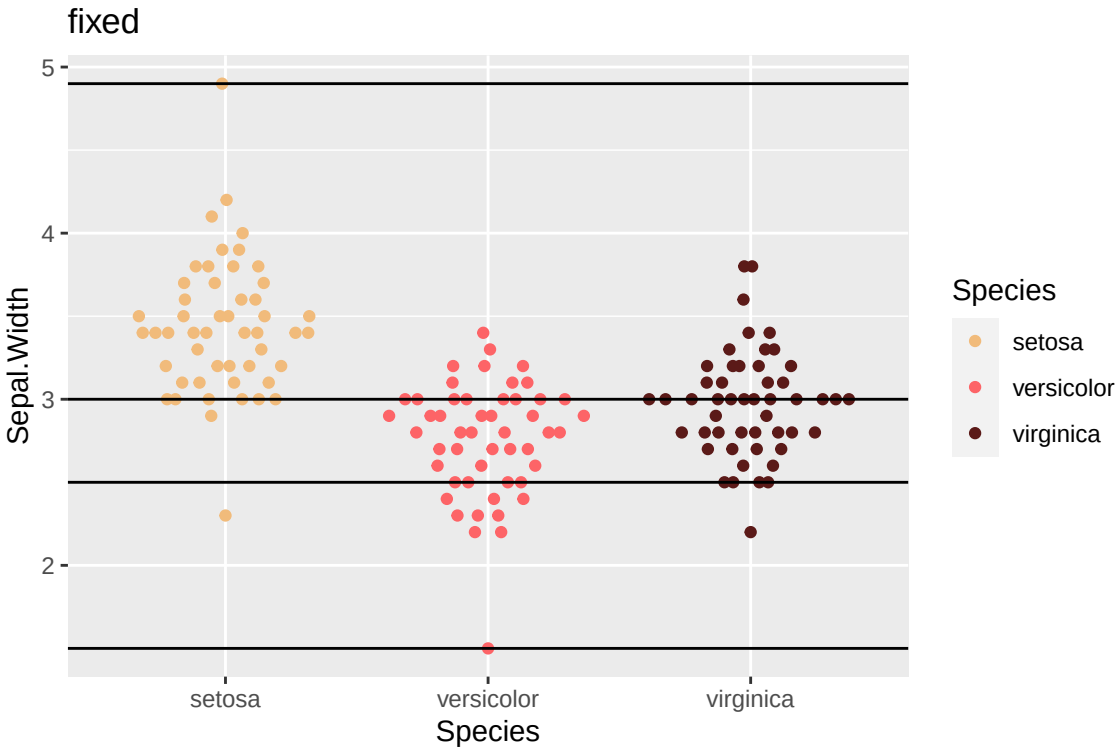

```
## Cases in matched pairs: 55.33 %
```






```
## Cases in matched pairs: 57.33 %
```






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

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.

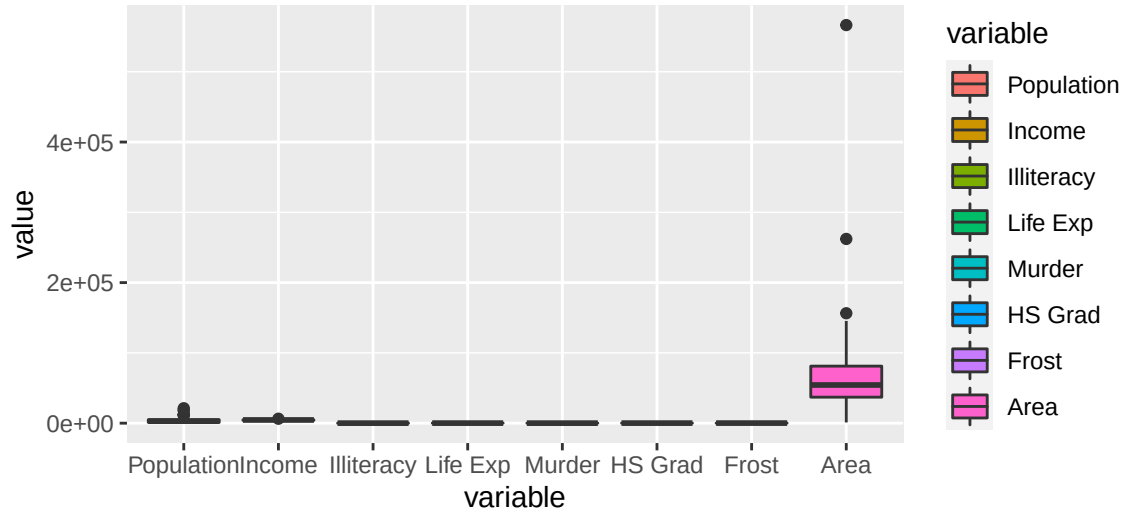


Tabela 1: Odchylenie standardowe i wspolczynnik zmiennosci dla zmienych

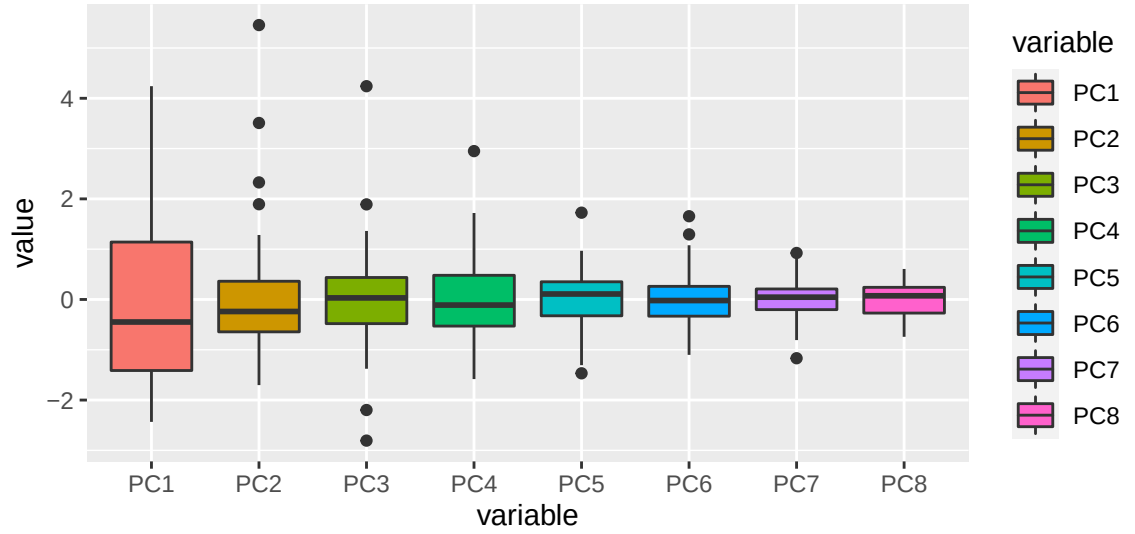
	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
Odchylenie standardowe	4464.491	614.470	0.610	1.342	3.692	8.077	51.981	85327.300
Wspolczynnik zmiennosci	1.051	0.139	0.521	0.019	0.500	0.152	0.498	1.206

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

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)
```



Przypatrzmy się teraz wektorom ładunków dla trzech pierwszych składowych głównych.

Tabela 2: Odchylenie standardowe i wspolczynnik zmiennosci dla zmienych

	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

```
# tutaj wektor ładunków - barplot
```

Wnioski — jeszcze się napisze . . .

Zbadajmy teraz jaka część wyjaśnionej wariancji odpowiada kolejnym składowym głównym.

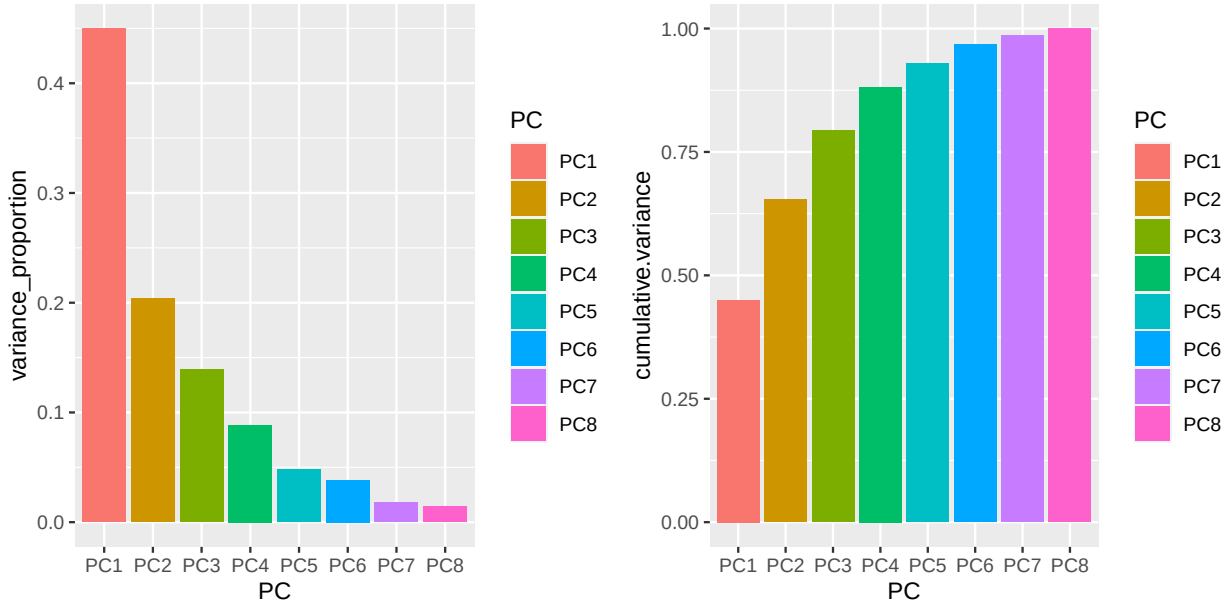


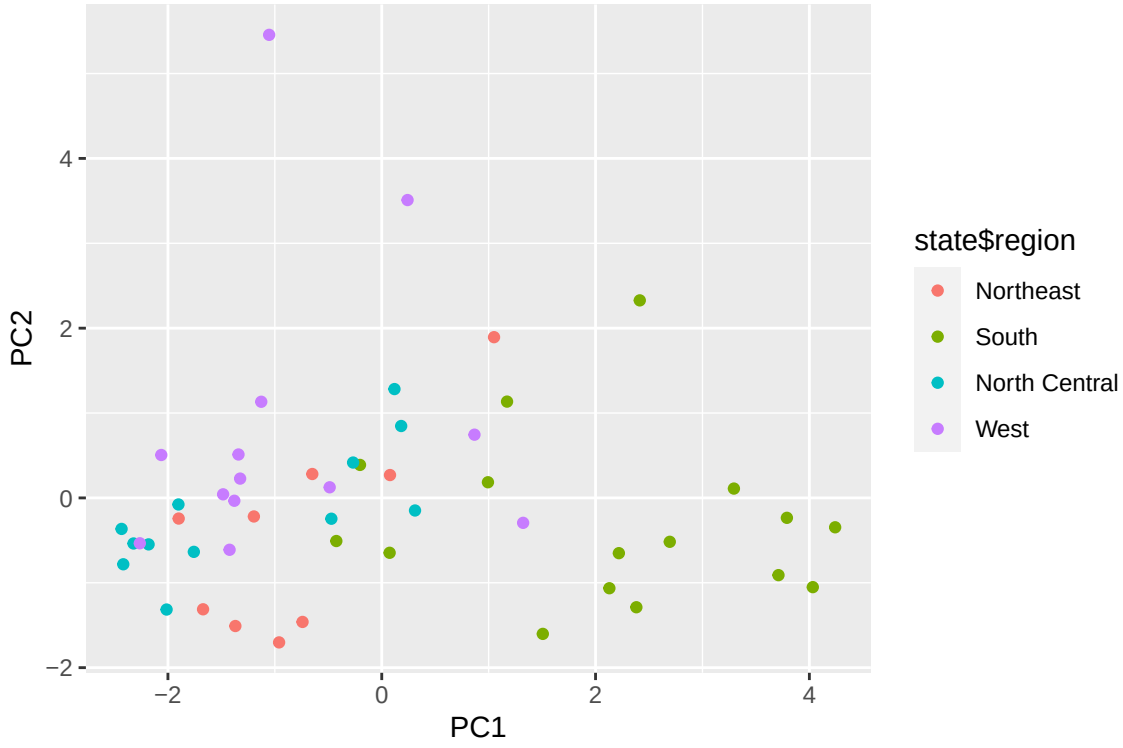
Tabela 3: Odchylenie standardowe i wspolczynnik zmiennosci dla zmienych

	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

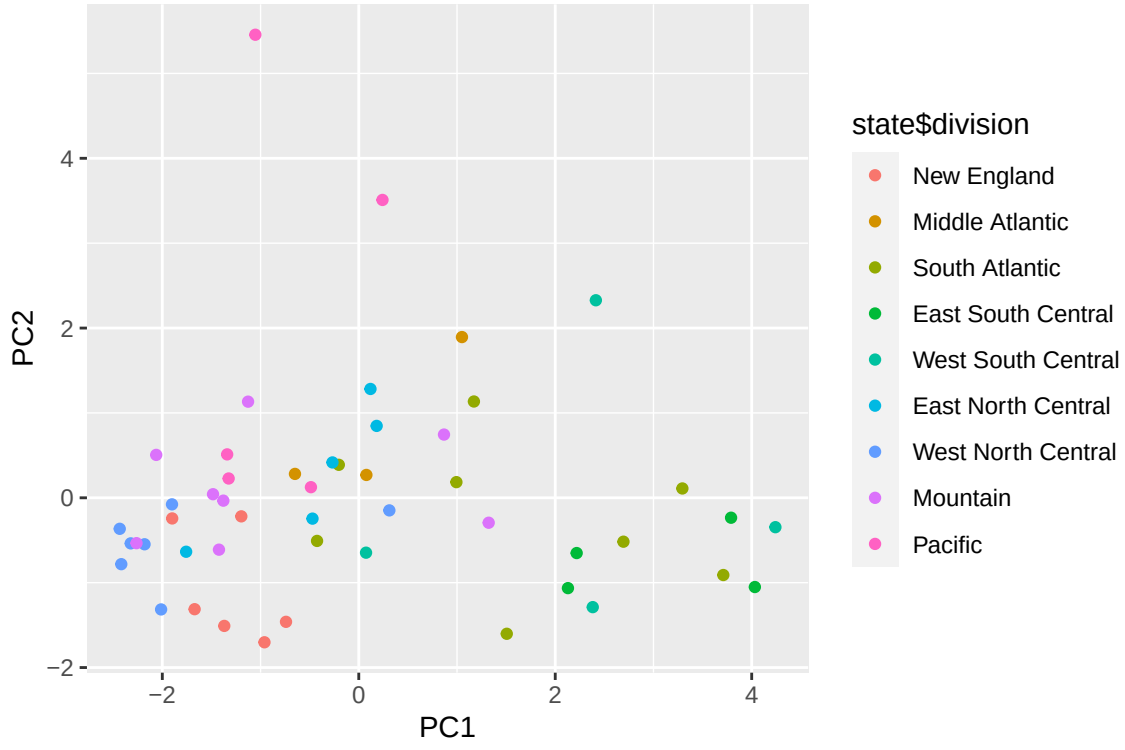
Wnioski — jeszcze się napisze ...

3.3 Wizualizacja danych

W tej części wygenerujemy wykresy rozrzutu 2d dla dwóch pierwszych składowych głów-



nych.



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

3.4 Korelacja zmiennych

Wygenerujmy teraz dwuwykres.

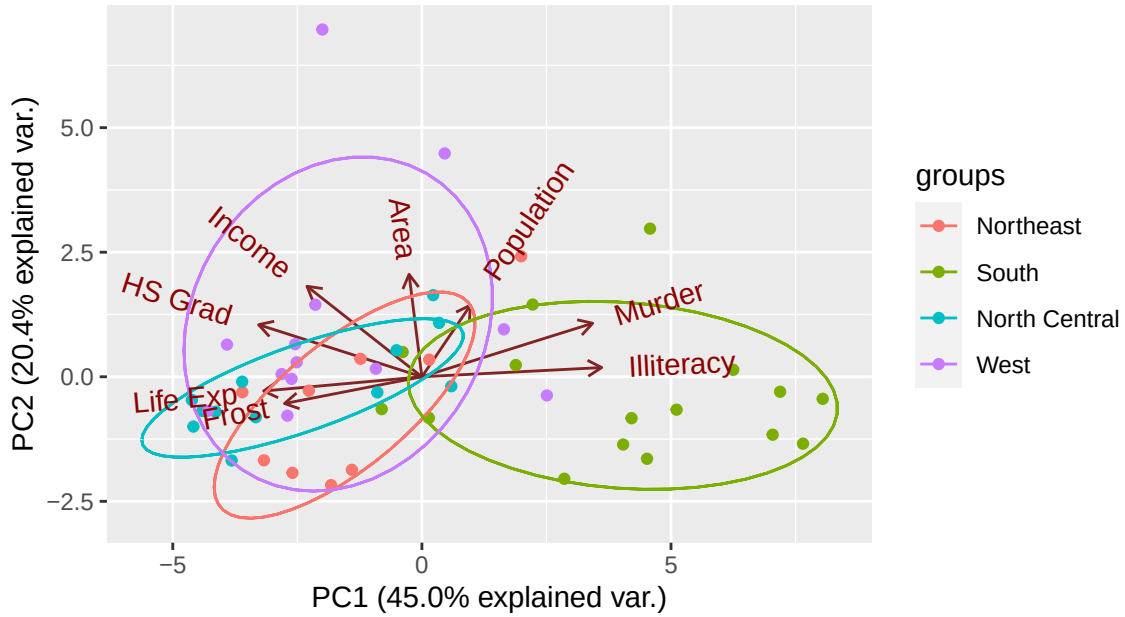


Tabela 4: Odchylenie standardowe i wspolczynnik zmiennosci dla zmienych

	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