Raport nr 2

Emilia Kowal [249716], Jakub Dworzański [249703]

26 kwietnia 2020

Spis treści

1 Dyskretyzacja cech ciągłych.

1.1 Krótki opis zagadnienia.

W tej sekcji przyjrzymy się działaniu różnych metod dyskretyzacji nienadzorowanej wykorzystując w tym celu zbiór danych *iris* z pakietu *datasets*, który zawiera zestaw pomiarów kwiatów oraz informację o gatunku irysa. Na podstawie otrzymanych wyników postaramy się przeprowadzić analizę skuteczności algorytmów.

Tabela 1: Opis danych

Zmienna	Typ zmiennej	Pierwsze wartości
Sepal.Length	double	5.1, 4.9, 4.7, 4.6
Sepal.Width	double	3.5, 3, 3.2, 3.1
Petal.Length	double	1.4, 1.4, 1.3, 1.5
Petal.Width	double	0.2, 0.2, 0.2, 0.2
Species	integer	setosa, setosa, setosa, setosa

1.2 Opis eksperymentów/analiz

Przedsięwzięcie rozpoczynamy od wyboru cech o najgorszych oraz najlepszych zdolnościach dyskryminacyjnych. Dla wybranych zmiennych stosujemy algorytmy:

- dyskretyzacja według równej szerokości przedziałów,
- dyskretyzacja według równej częstości,
- dyskretyzacja oparta na algorytmie k-means,
- dyskretyzacja z przedziałami wyznaczonymi ręcznie.

Następnie dokonujemy porównania wyników z rzeczywistymi etykietami klas. Zbadamy również wpływ obserwacji odstających na powyższe algorytmy.

1.3 Wyniki

1.3.1 Przygotowanie danych. Podstawowe informacje o danych.

```
data(iris)
attach(iris)
```

Zbiór danych zawiera 150 obserwacji dotyczących 3 gatunków irysa: setosa, versicolor oraz virginica. Po wczytaniu danych do przestrzeni roboczej sprawdzamy poprawność typów zmiennych.

Na podstawie tabeli 1 widzimy, że wszystkie zmienne zostały wczytane poprawnie. Następnie sprawdzamy, czy w zbiorze znajdują się wartości brakujące.

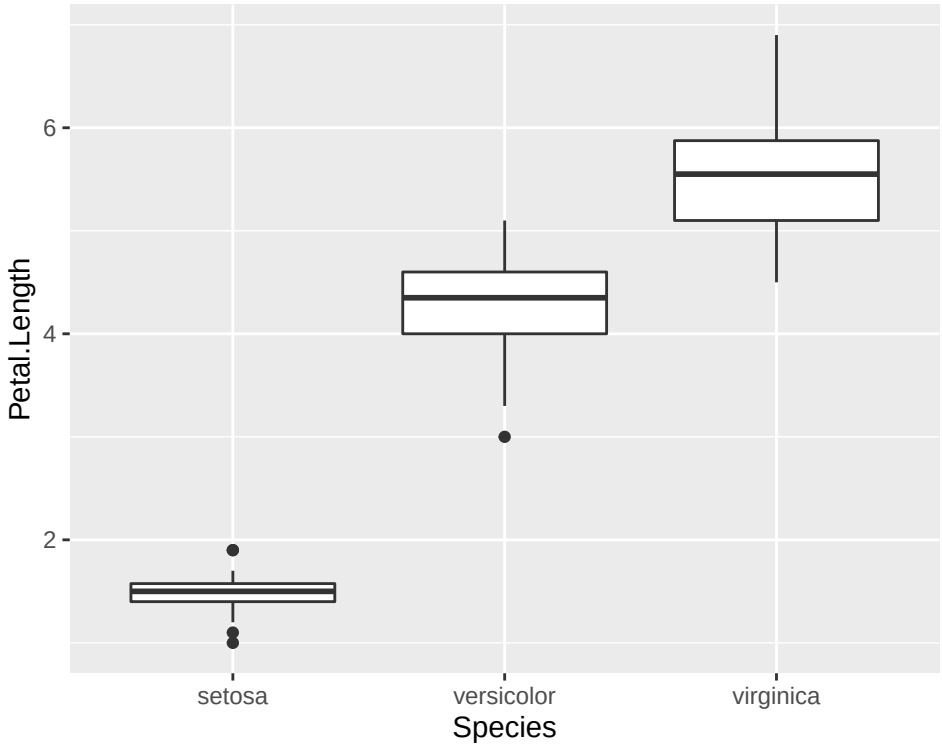
```
sum(is.na(iris))

## [1] 0
```

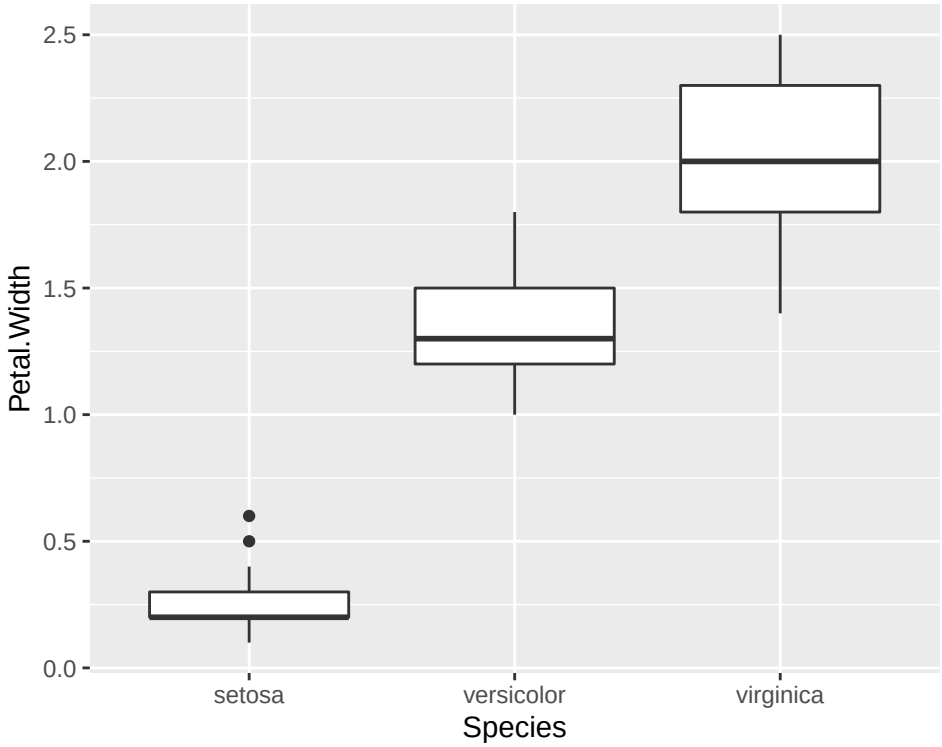
Widzimy, że w zbiorze *iris* nie ma wartości brakujących.

1.3.2 Analiza zdolności dyskryminacyjnych cech.

W celu identyfikacji cech o najgorszych oraz najlepszych zdolnościach dyskryminacyjnych prezetujemy dane na wykresach pudełkowych i wykresach rozrzutu.

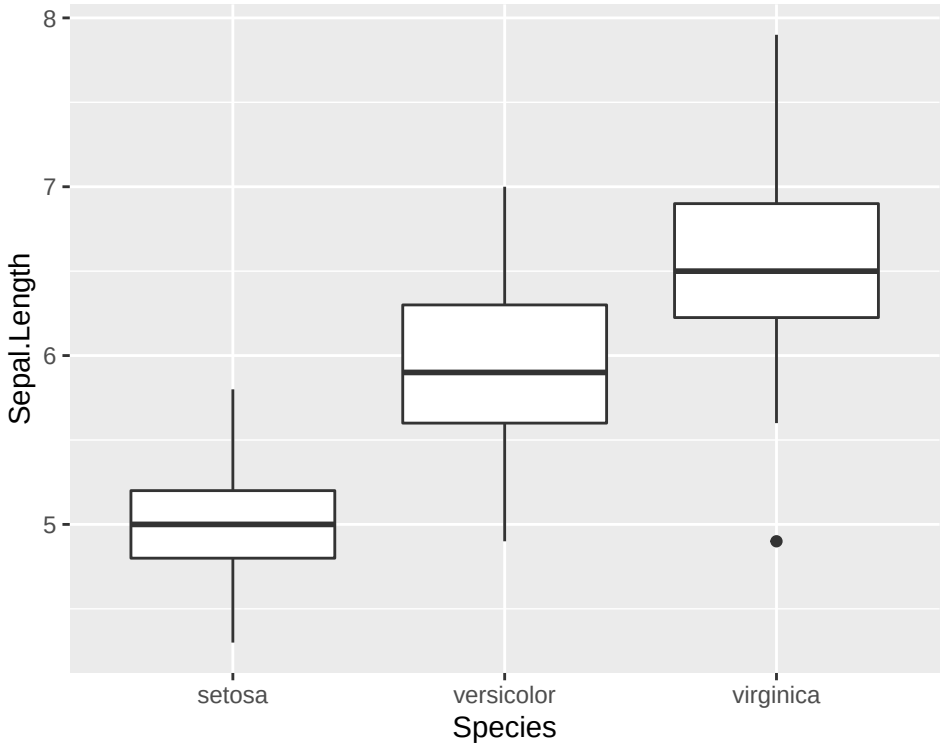


Rysunek 1: Wykres pudełkowy dla zmiennej Petal.Length

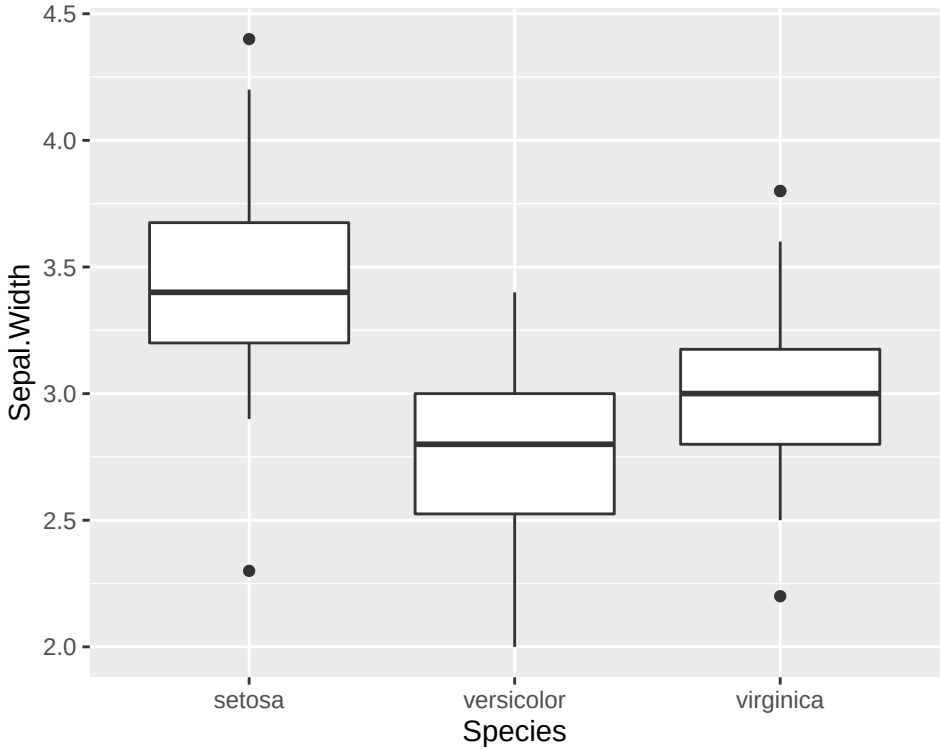


Rysunek 2: Wykres pudełkowy dla zmiennej Petal.Width

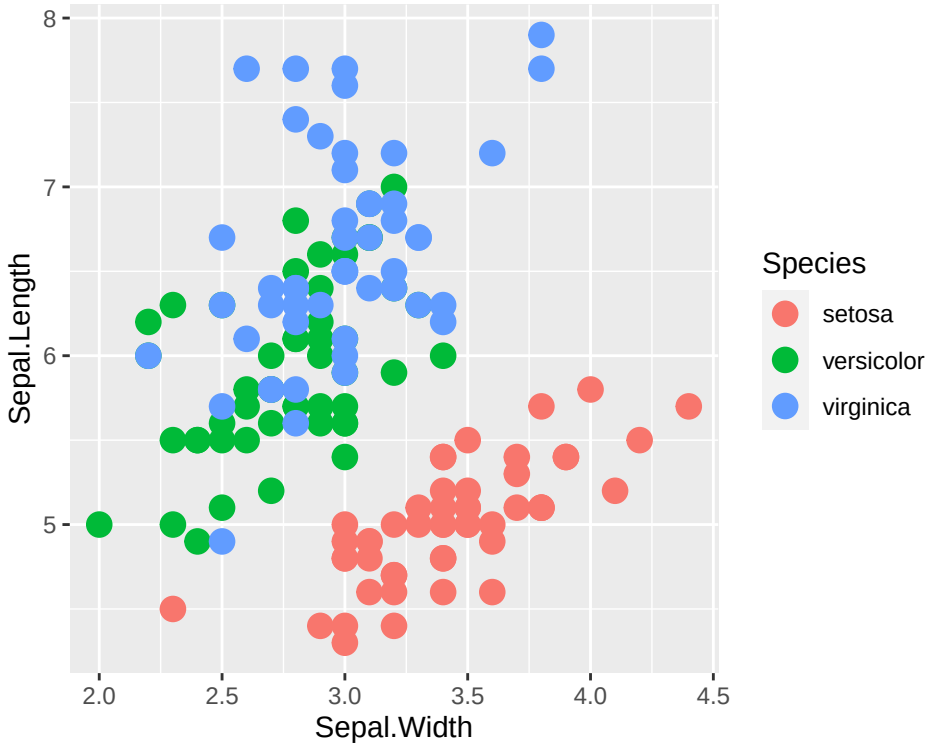
Rysunek 3: Wykres rozrzutu dla zmiennych Petal.Width i Petal.Length



Rysunek 4: Wykres pudełkowy dla zmiennej Sepal.Length



Rysunek 5: Wykres pudełkowy dla zmiennej Sepal.Width



Rysunek 6: Wykres rorzutu dla zmiennych Sepal.Width i Sepal.Length

Na podstawie analizy wykresów 1, 2 i 3 możemy stwierdzić, iż cechy *Petal.Length* oraz *Petal.Width* wykazują najlepsze zdolności dyskryminacyjne, natomiast cechy *Sepal.Length* oraz

Tabela 2: Tabela kontyngencji dla dyskretyzacji Petal.Width metodą equal width.

	setosa	versicolor	virginica
[0.1,0.9)	50	0	0
[0.9,1.7)	0	48	4
[1.7,2.5]	0	2	46

Tabela 3: Tabela kontyngencji dla dyskretyzacji Petal.Width metodą fixed.

	setosa	versicolor	virginica
[-Inf,0.75)	50	0	0
[0.75,1.65)	0	48	4
[1.65, Inf]	0	2	46

Sepal.Width (wykresy: 4, 5 i 6) posiadają najgorsze zdolności dyskryminacyjne.
Algorytmy dyskretyzacji zastosujemy dla zmiennych Petal.Width oraz Sepal.Width.

1.3.3 Porównanie nienadzorowanych metod dyskretyzacji.

Dla zmiennej *Petal.Width* otrzymujemy następujące wyniki:

- dla dyskretyzacji według równych przedziałów (tabela: 2), według przedziałów zadanych ręcznie (tabela: 3) oraz wykorzystującej algorytm k-means (tabela: 4) dostajemy wynik o poziomie zgodności 96%,
- dla dyskretyzacji według równych częstości (tabela: 5) otrzymujemy najgorszy wynik, współczynnik zgodności wynosi 94.67%.

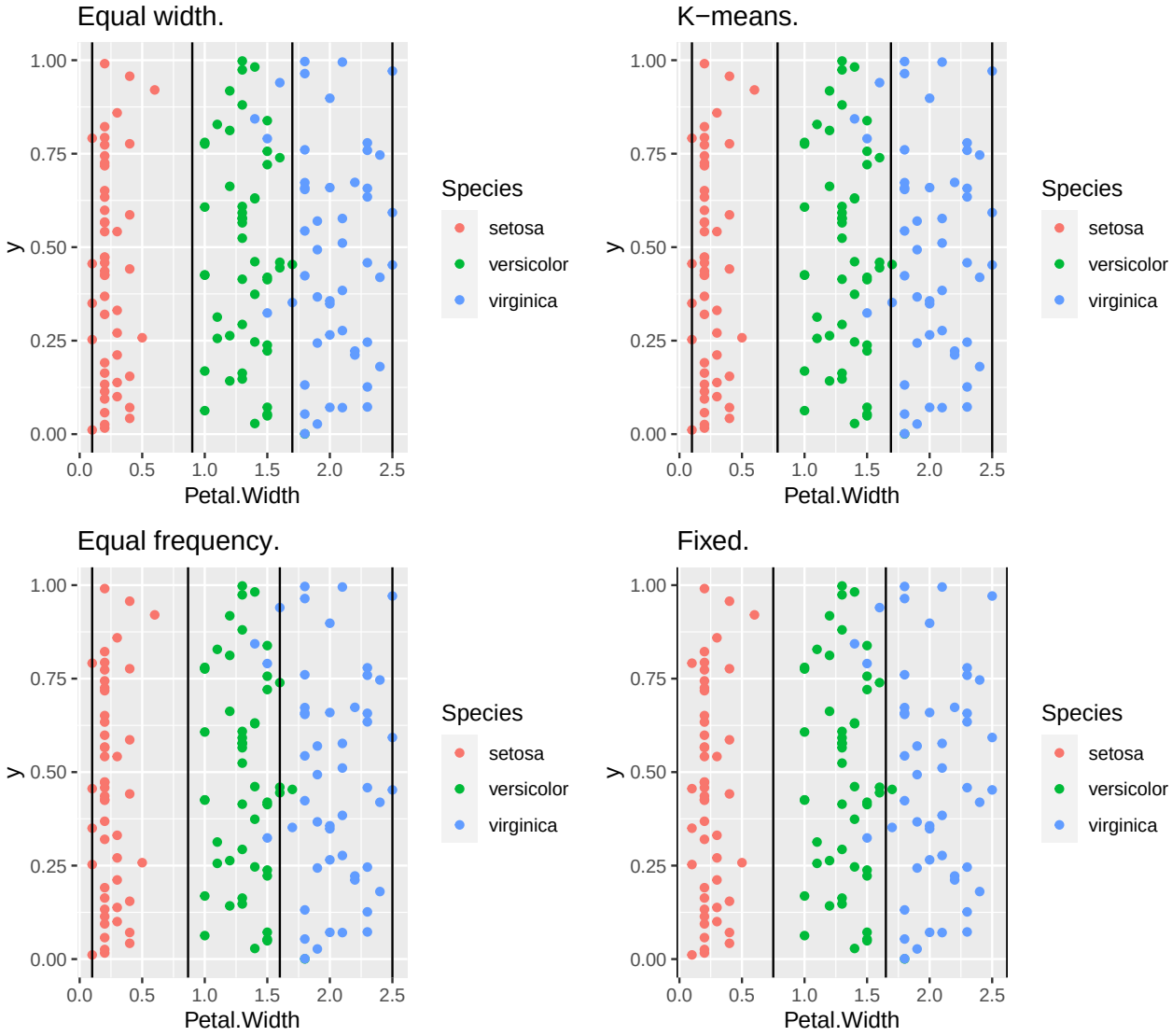
Porównujemy, na wykresach rozrzutu, wyniki poszczególnych algorytmów dyskretyzacji dla zmiennej *Petal.Width*.

Tabela 4: Tabela kontyngencji dla dyskretyzacji Petal.Width metodą k-means.

	setosa	versicolor	virginica
[0.1,0.785)	50	0	0
[0.785,1.69)	0	48	4
[1.69,2.5]	0	2	46

Tabela 5: Tabela kontyngencji dla dyskretyzacji Petal.Width metodą equal frequency.

	setosa	versicolor	virginica
[-Inf,0.75)	50	0	0
[0.75,1.65)	0	48	4
[1.65, Inf]	0	2	46



Rysunek 7: Wyniki dyskretyzacji dla zmiennej Petal.Length

Widzimy na wykresach rozrzutu, że dla zmiennej *Petal.Width* o najlepszych zdolnościach dyskryminacyjnych, przedziały wyznaczone przez zastosowane algorytmy, pozwalają nam z dużym przybliżeniem określić, do której z klas należy dana próba.

Dla zmiennej *Sepal.Width*, o słabych zdolnościach dyskryminacyjnych, otrzymujemy następujące wyniki:

- dla dyskretyzacji wykorzystującej algorytm k-means (tabela: 8) dostajemy najwyższy współczynnik zgodności-56%,

Tabela 6: Tabela kontyngencji dla dyskretyzacji Sepal.Width metodą equal width.

	setosa	versicolor	virginica
[2,2.8)	1	27	19
[2.8,3.6)	36	23	29
[3.6,4.4]	13	0	2

Tabela 7: Tabela kontyngencji dla dyskretyzacji Sepal.Width metodą fixed.

	setosa	versicolor	virginica
[-Inf,2.75)	1	21	11
[2.75,3.45)	27	29	36
[3.45, Inf]	22	0	3

Tabela 8: Tabela kontyngencji dla dyskretyzacji Sepal.Width metodą k-means.

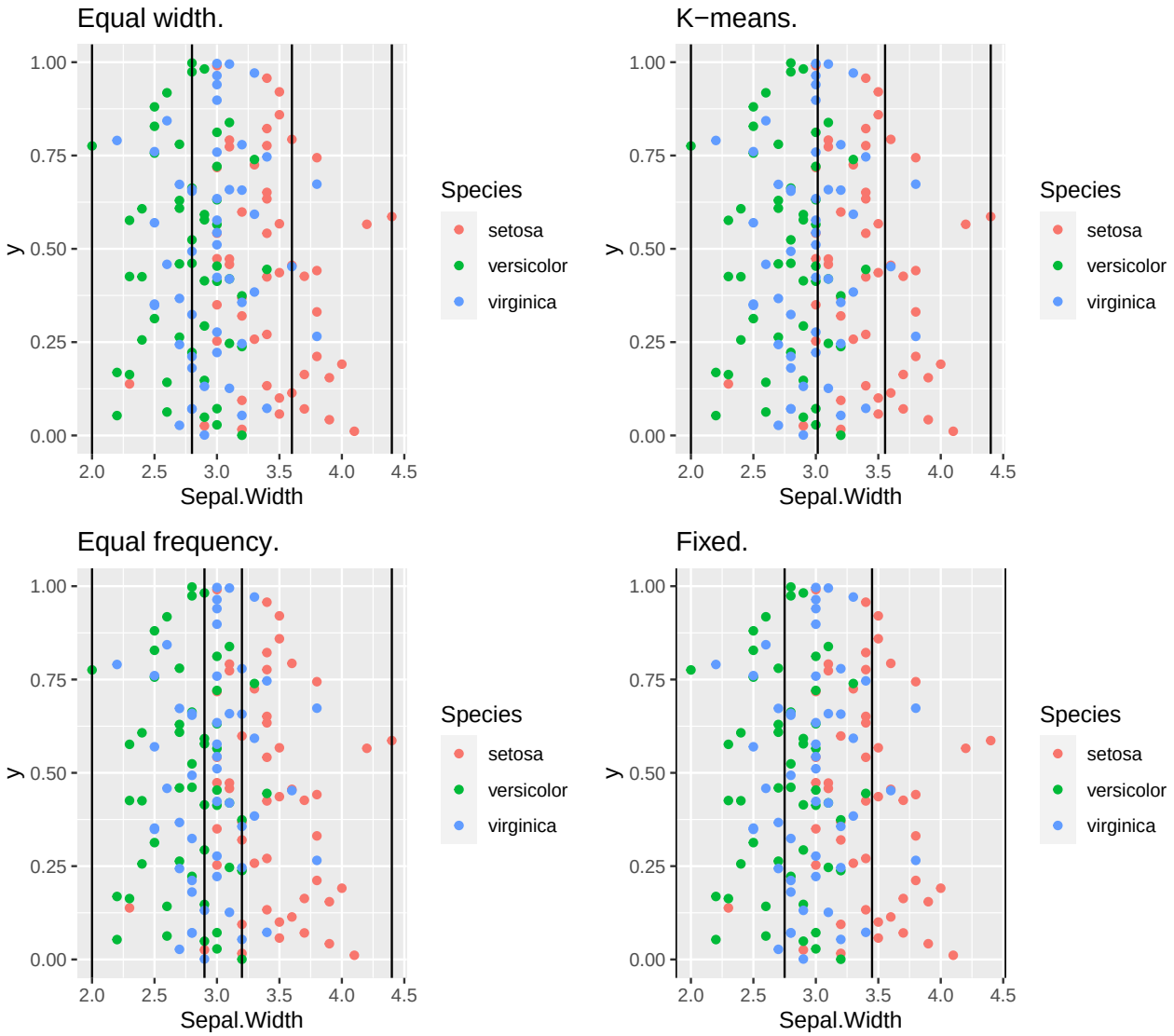
	setosa	versicolor	virginica
[2,3.02)	8	42	33
[3.02,3.55)	26	8	14
[3.55,4.4]	16	0	3

Tabela 9: Tabela kontyngencji dla dyskretyzacji Sepal.Width metodą equal frequency.

	setosa	versicolor	virginica
[-Inf,2.75)	1	21	11
[2.75,3.45)	27	29	36
[3.45, Inf]	22	0	3

- dla dyskretyzacji według równych częstości (tabela: 9) otrzymujemy zgodność na poziomie 55.33%,
- dla dyskretyzacji metodą fixed (tabela: 7) dostajemy zgodność: 52.67%,
- najgorszy wynik obserwujemy dla dyskretyzacji według równych przedziałów (tabela 6. Współczynnik zgodności wynosi 50.67%.

Porównujemy, na wykresach rozrzutu, wyniki poszczególnych algorytmów dyskretyzacji dla zmiennej *Sepal.Width*.



Rysunek 8: Wyniki dyskretyzacji dla zmiennej Sepal.Length

Również na wykresie rozrzutu dla dyskretyzacji według równych przedziałów można dostrzec, że algorytm, w porównaniu do pozostałych trzech, poradził sobie najgorzej ze zmienną *Sepal.Width*.

Tabela 10: Tabela kontyngencji dla dyskretyzacji metodą equal frequency zmodyfikowanej Petal.Width.

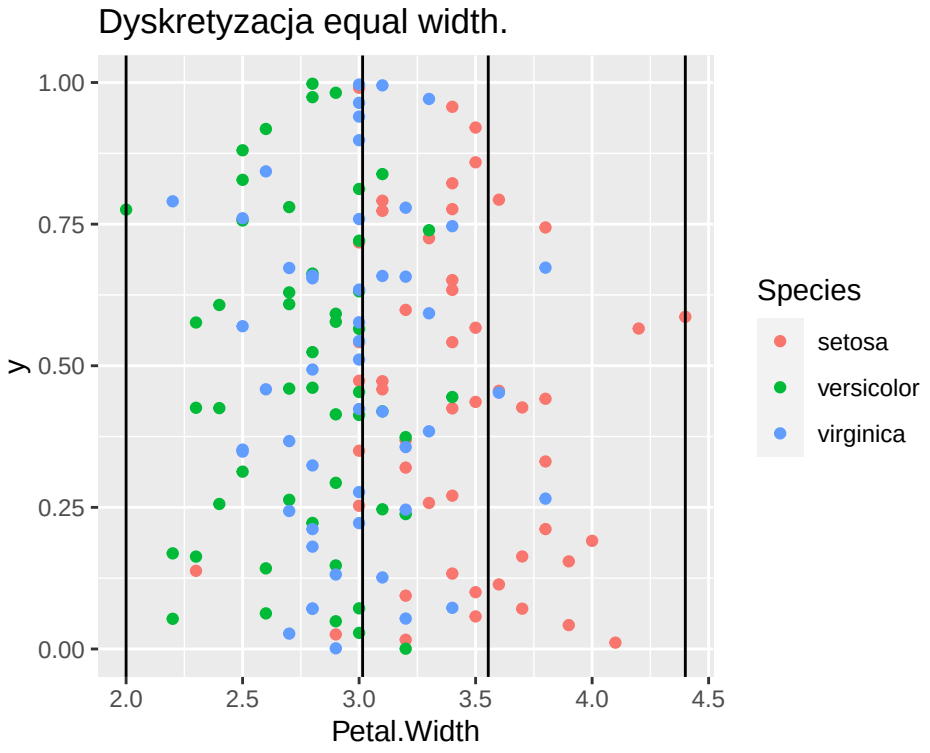
	setosa	versicolor	virginica
[-2.9,-0.1)	1	0	0
[-0.1,2.7)	49	50	49
[2.7,5.5]	0	0	1

1.3.4 Wpływ wartości odstających na metody dyskretyzacji.

Analizę zmiennej Petal.Width powtarzamy, zastępując wartość najmniejszą oraz największą cechy, wartościami odstającymi.

```
d1 <- Petal.Width
d1[which.min(d1)] <- min(d1) - 2*IQR(d1)
d1[which.max(d1)] <- max(d1) + 2*IQR(d1)
```

Obecność wartości odstających znacząco wpływa na dyskretyzację według równych przedziałów. Dla transformacji przeprowadzonej na zmodyfikowanej zmiennej Petal.Width otrzymujemy zgodność 34.67% (tabela: 10), czyli aż o 61,33 punkta procentowego mniejszą od wyniku dla Petal.Width bez wartości odstających.



Rysunek 9: Dyskretyzacja equal width dla Petal.Width z wartościami odstającymi.

Widzimy na wykresie, iż wartości krańców przedziałów zostały zdeterminowane przez wartości odstające zmiennej *Petal.Width*.

Tabela 11: Przykładowe dane

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708
Alaska	365	6315	1.5	69.31	11.3	66.7	152	566432
Arizona	2212	4530	1.8	70.55	7.8	58.1	15	113417
Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945
California	21198	5114	1.1	71.71	10.3	62.6	20	156361
Colorado	2541	4884	0.7	72.06	6.8	63.9	166	103766

1.4 Podsumowanie

Przeprowadzone eksperymenty pozwalają nam dostrzec, iż niezwykle ważnym dla przeprowadzenia dyskretyzacji, jest dokładna analiza zdolności dyskryminacyjnych zmiennych. Wnioskujemy, iż dla rozkładów ciężkoogonowych nie stosujemy dyskretyzacji opartej na równych przedziałach, wrażliwej na wartości odstające. Dla zbioru iris, na podstawie cechy *Petal.Width*, jesteśmy w stanie określić z dużą dokładnością, do której grupy przynależy dana próba.

2 Analiza składowych głównych (Principal Component Analysis (PCA))

2.1 Krótki opis zagadnienia

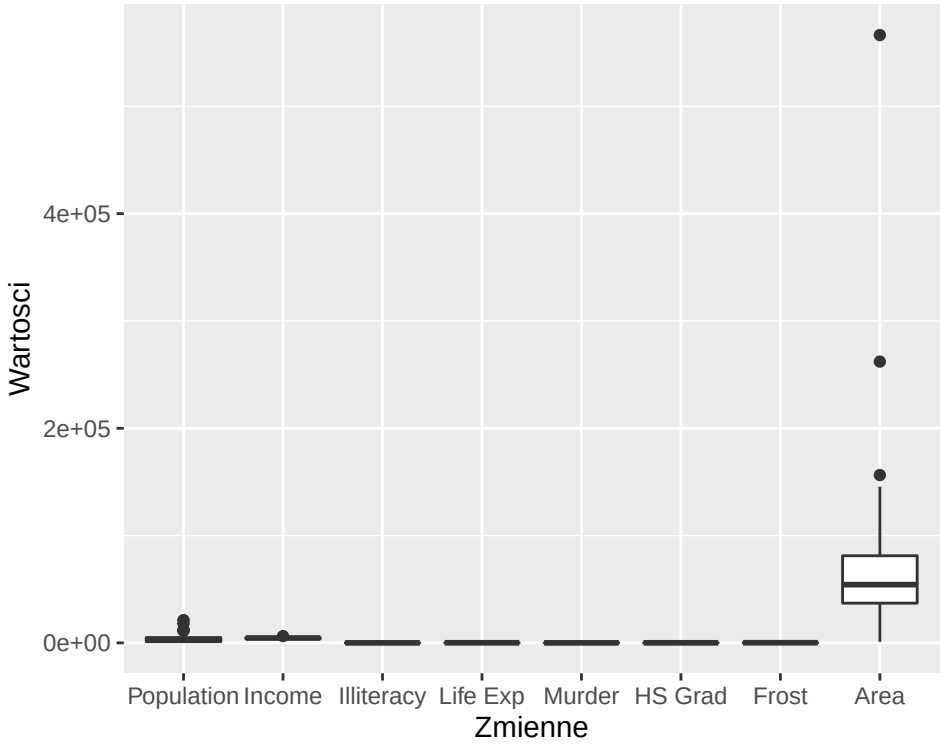
Celem tego ćwiczenia będzie przede wszystkim zaznajomienie się z algorytmem PCA oraz ocena jego przydatności. Sprawdzimy jakie możliwości redukcji wymiaru dostarcza i jak wiele informacji jesteśmy w stanie, dzięki niemu, zachować po przekształceniu danych do mniejszych wymiarów.

2.2 Opis eksperymentów/analiz

Posłużymy się zbiorem `state.x77` z pakietu `datasets`. Dane te zawierają podstawowe informacje o każdym ze stanów. Po przekształceniu danych przy pomocy PCA, przeanalizujemy je w nowej postaci. Na początku sprawdzimy jak wiele całkowitej wariancji wyjaśniają poszczególne główne składowe, a następnie przeanalizujemy wizualnie dane w nowej postaci. Poza tym, posługując się wektorami ładunków, dwuwykresem oraz macierzą korelacji postaramy się zdobyć więcej informacji o zależnościach między zmiennymi.

2.3 Wyniki

```
head(dane)
ggplot(stack(dane), aes(x=ind, y=values)) + geom_boxplot()
```



Rysunek 10: Wykresy pudełkowe zmienności poszczególnych zmiennych ze zbiory danych.

W tabeli 11 widzimy przykładowe dane. Na podstawie wykresu 10 możemy stwierdzić, że przed zastosowaniem PCA, będziemy musieli dokonać standaryzacji danych, ponieważ zmienna `Area` dominuje wariancję wszystkich zmiennych.

Dlatego też standaryzujemy dane i wyznaczamy składowe główne.

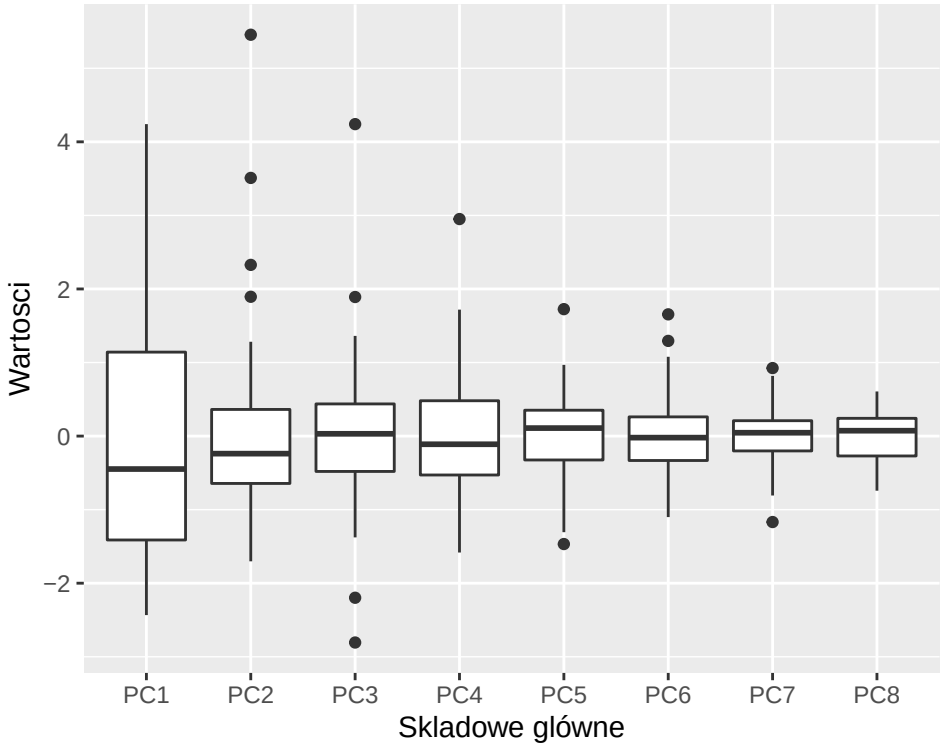
```
dane.pca <- prcomp(dane, retx=T, center=T, scale.=T)
```

Teraz możemy zbadać rozrzut składowych głównych.

```
pc <- data.frame(dane.pca$x)
ggplot(stack(pc), aes(x=ind, y=values)) + geom_boxplot() +
  labs(x="Składowe główne", y="Wartości")
```

Tabela 12: Wektory ładunków dla składowych głównych PC1, PC2, PC3, PC4.

	PC1	PC2	PC3	PC4
Population	0.1264281	0.4108742	-0.6563255	-0.4093856
Income	-0.2988299	0.5189788	-0.1003592	-0.0884466
Illiteracy	0.4676692	0.0529687	0.0708985	0.3528280
Life Exp	-0.4116104	-0.0816561	-0.3599330	0.4425633
Murder	0.4442567	0.3069493	0.1084675	-0.1656002
HS Grad	-0.4246844	0.2987666	0.0497085	0.2315741
Frost	-0.3574124	-0.1535841	0.3871145	-0.6186512
Area	-0.0333846	0.5876245	0.5103850	0.2011255



Rysunek 11: Analiza rozproszenia składowych głównych.

Korzystając z wykresu 11, możemy zauważyć, że dla zmiennej PC1 rozstęp międzykwartylowy ma największą wartość.

Dla pierwszych czterech głównych składowych, przeanalizujemy wektory ładunków.

```
dane.pca$rotation[,1:4]
```

W tabeli 12 możemy zobaczyć, że:

- I wektor ładunków przypisuje w przybliżeniu jednakową wagę zmiennym: `Illiteracy`, `Life Exp`, `Murder` oraz `HS Grad`, zatem PC1 można interpretować jako wskaźnik poziomu edukacji oraz długości życia i popełnionych morderstw.
- II wektor ładunków przypisuje największe wagi dla zmiennych `Area`, `Income`, `Population`,

Tabela 13: Udział głównych składowych w całkowitej wariancji.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Odchylenie standardowe	1.90	1.28	1.05	0.84	0.62	0.55	0.38	0.34
Część całkowitej wariancji	0.45	0.20	0.14	0.09	0.05	0.04	0.02	0.01
Skumulowana część całkowitej wariancji	0.45	0.65	0.79	0.88	0.93	0.97	0.99	1.00

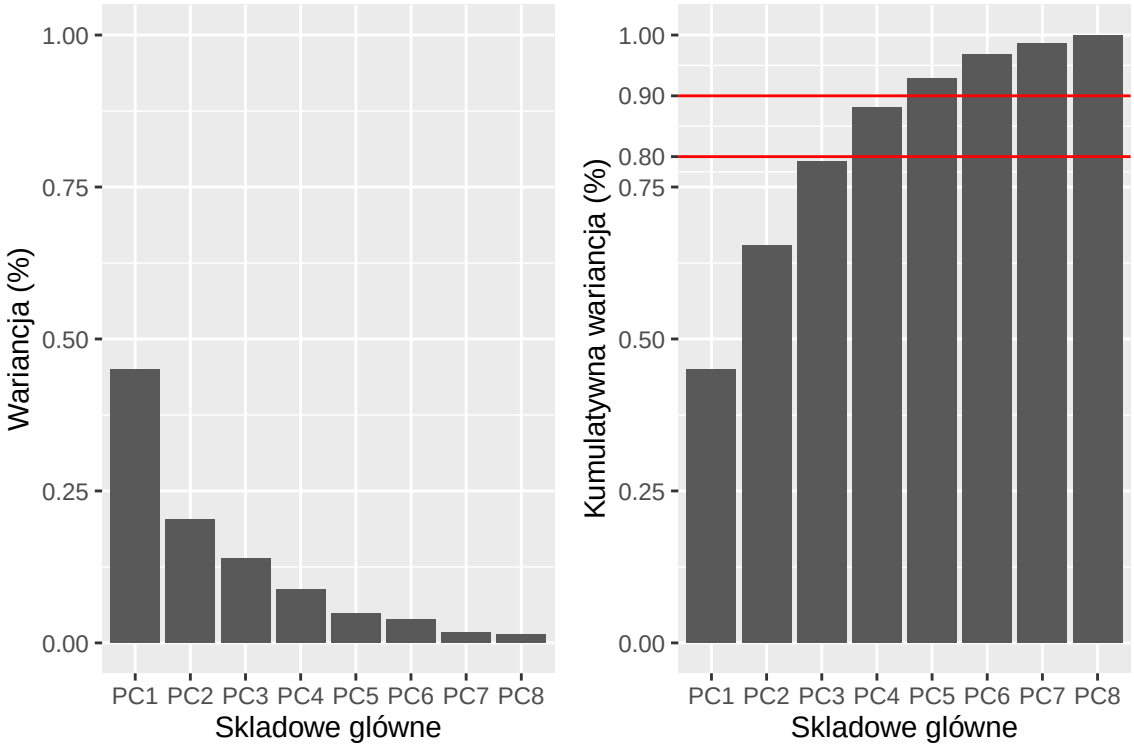
czyli PC2 można uznać za wskaźnik zaludnienia oraz wysokości zarobków na danym obszarze.

- III wektor ładunków przypisuje największą wagę zmiennej `Population` oraz `Area`, zatem opisuje poziom zaludnienia.
- IV wektor ładunków przypisuje największą wagę zmiennej `Frost`, zatem wskazuje na częstość występowania mrozów w danym stanie

Kiedy już mamy wstępne informacje, możemy przeanalizować, jaki procent wyjaśnionej wariancji odpowiada poszczególnym składowym głównym.

```
summary(dane.pca)
wariancja <- (dane.pca$sdev ^2)/sum(dane.pca$sdev^2)
kum.wariancja <- cumsum(wariancja)
df_zmiennosc <- data.frame(wariancja, kum.wariancja, names(pc))

ggplot(df_zmiennosc, aes(x=names.pc., y=wariancja)) +
  geom_bar(stat='identity')
ggplot(df_zmiennosc, aes(x=names.pc., y=kum.wariancja)) +
  geom_bar(stat='identity')
```

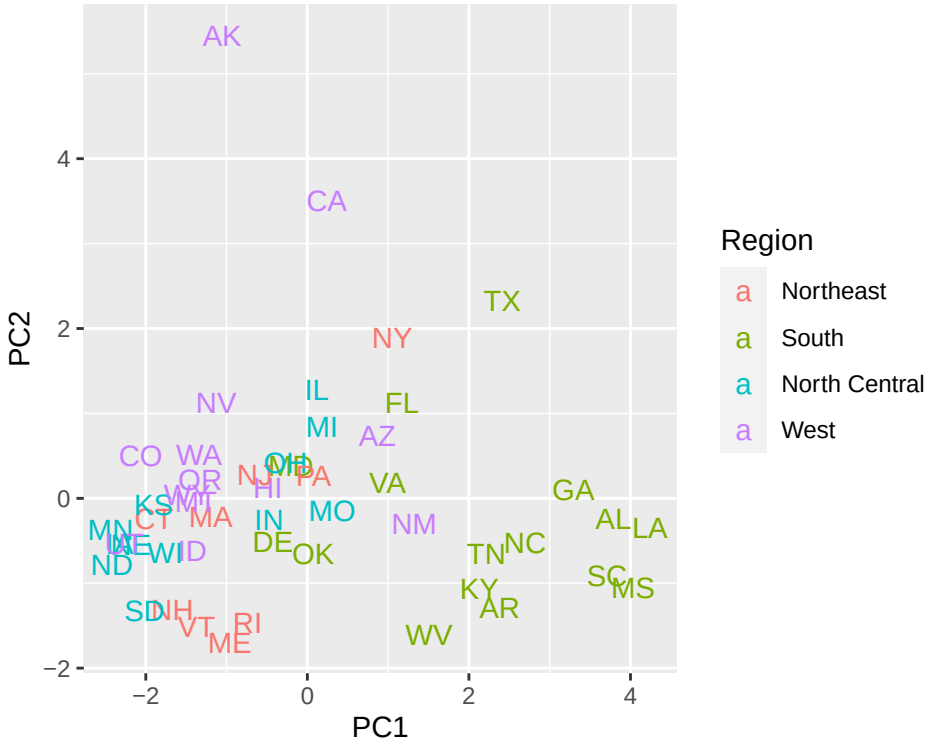


Rysunek 12: Procent zmienności oraz kumulatywnej zmienności dla poszczególnych składowych.

W tabeli 13 oraz na wykresie 12 widzimy jaki procent całkowitej zmienności odpowiada poszczególnym składowym. Ponadto widzimy, że do wyjaśnienia ponad 80% wariancji wystarczą cztery pierwsze główne składowe, a pierwsze pięć składowych wyjaśnia ponad 90% całkowitej wariancji.

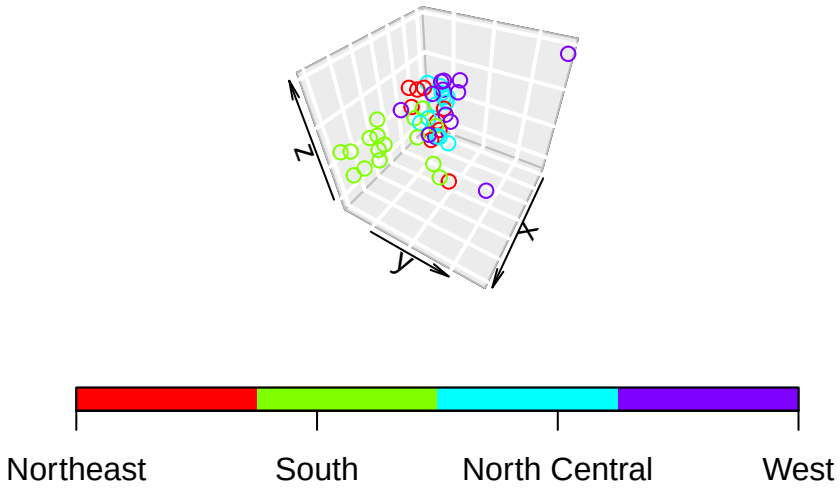
Teraz sprawdzimy, jak dobrze działa PCA poprzez wizualizację danych, czyli dla pierwszych trzech składowych głównych.

```
dane.PCA <- data.frame(dane.pca$x)
ggplot(dane.PCA, aes(x=PC1, y=PC2, col=state.region, label=state.abb)) +
  geom_text()
```



Rysunek 13: Wykres rozrzutu dwóch pierwszych składowych głównych.

```
attach(dane.PCA)
points3D(PC1, PC2, PC3)
```



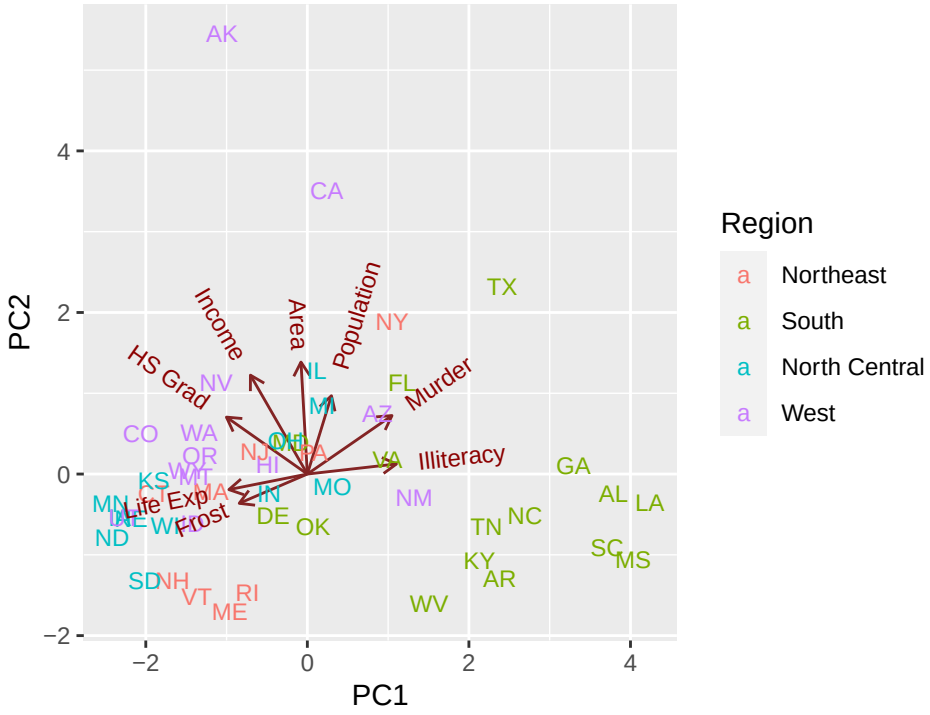
Rysunek 14: Wykres rozrzutu trzech pierwszych składowych głównych.

Na podstawie wykresów 13 i 14, możemy powiedzieć, że wśród danych poza wartościami odstającymi od reszty(przede wszystkim Alaska, ale też California, Texas, Nowy York i Florida), możemy wyszczególnić dwie grupy. Mniejsza z nich, to dość jednolita grupa, która składa się z większości południowych stanów. Reszta stanów zdaje się układać w dość zróżnicowaną grupę.

Widzimy również, że po przekształceniu otrzymaliśmy kilka stanów odstających od pozostałych. Przyczyny odseparowania Alaski można szukać między innymi w nieporównywalnie dużej powierzchni tego stanu.

Zbadamy dokładniej przedstawienie zmiennych w dwuwymiarowej przestrzeni.

```
ggbiplot(dane.pca, groups=state.region, labels=state.abb)
```

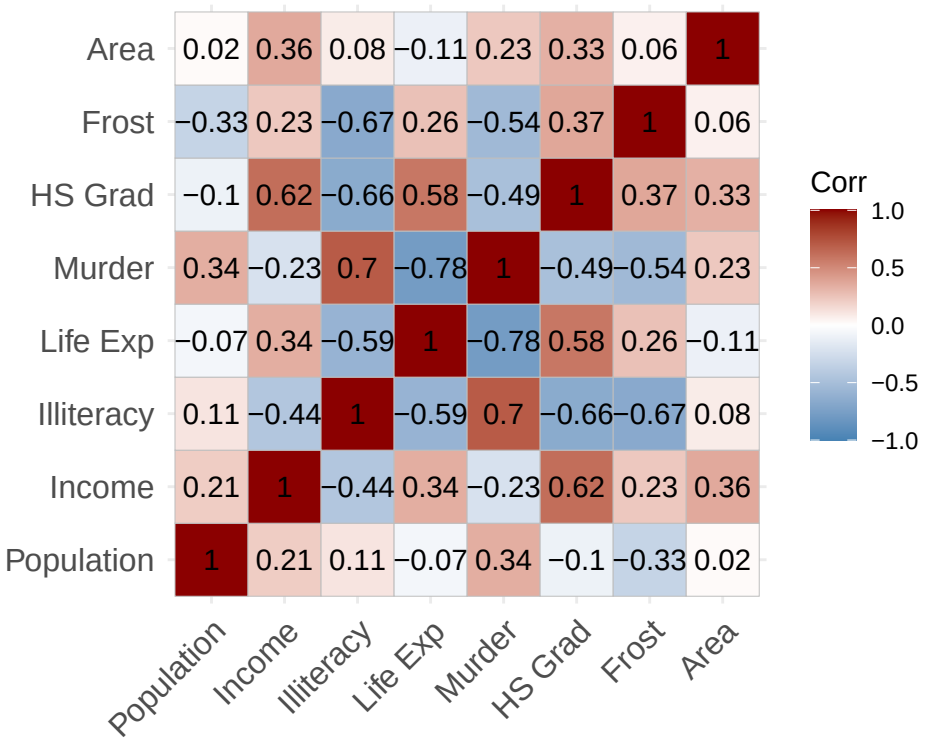


Rysunek 15: Dwuwykres dla zbioru danych po przekształceniu PCA.

Istotnie, korzystając z wykresu 15, widzimy, że wektor reprezentujący wzrost zmiennej `Area` jest skierowany w stronę Alaski.

Teraz postaramy się zbadać korelację pomiędzy zmiennymi na podstawie dwuwykresu oraz macierzy korelacji.

```
ggcorrplot(cor(dane))
```



Rysunek 16: Macierz korelacji

Na dwuwykresie (Rys. 15), możemy zauważyć, że wskaźnik morderstw może być skorelowany z analfabetyzmem. Ponadto, co wydaje się być dość intuicyjne, wskaźnik morderstw wygląda na ujemnie skorelowany z oczekiwaną długością życia. Ponadto, z wykresu wynika, że wskaźnik mroźnych dni w ciągu roku jest skorelowany z oczekiwaną długością życia i negatywnie skorelowany ze wskaźnikiem morderstw. Widać również, że populacja jest skorelowana z powierzchnią stanu, a procent wyższego wykształcenia jest skorelowany z przychodem.

Korzystając z macierzy korelacji (Rys. 16), możemy zauważyć, że większość z naszych przewidywań jest słuszna. Należy jednak zaznaczyć, że współczynnik korelacji między ilością mroźnych dni, a oczekiwaną długością życia, jest niższy, niż moglibyśmy tego oczekiwać, po spojrzeniu na dwuwykres. Ponadto, widzimy, że wbrew dwuwykresowi, współczynnik korelacji pomiędzy powierzchnią stanu, a populacją jest bardzo bliski 0.

2.4 Podsumowanie

Widzimy, że algorytm PCA pozwolił na przystępne i łatwe w wizaualizacji przedstawienie zbioru danych. Ponadto, zobaczyliśmy, że już trzy pierwsze składowe główne wyjaśniają 79% całkowitej wariancji. Należy pamiętać, że do przeprowadzenie PCA, konieczna było standaryzacja danych. Ciekawym wynikiem jest to, że w danych został odzwierciedlony podział północ-południe. Warto również zapamiętać, że podczas analizy otrzymaliśmy kilka charakterystycznych stanów, które reprezentują szczególne cechy (przede wszystkim Alaska).

Dlatego też, możemy ocenić pozytywnie przydatność algorytmu PCA do analizy danych. Znaczna redukcja wymiaru pozwoliła jednocześnie zachować dość dużo informacji o danych.

Tabela 14: Przykładowe dane

Pclass	Sex	Age	sibsp	Parch	Fare	Embarked	Survived
3	M	22	1	0	7.2500	2	N
1	F	38	1	0	71.2833	0	Y
3	F	26	0	0	7.9250	2	Y
1	F	35	1	0	53.1000	2	Y
3	M	35	0	0	8.0500	2	N
3	M	28	0	0	8.4583	1	N

3 Skalowanie wielowymiarowe (Multidimensional Scaling (MDS))

3.1 Krótki opis zagadnienia

W tej części będziemy się zajmowali MDS, czyli skalowaniem wielowymiarowym. Zdecydowaliśmy się na zbadanie skalowania niemetrycznego, które jest wariantem MDS.

3.2 Opis eksperymentów/analiz

Będziemy sprawdzali jakość odwzorowania MDS. W tym celu zbadamy, jak zmieniają się wartości funkcji `STRESS` oraz diagramy Shepparda dla różnych wymiarów docelowej przestrzeni. Do badań wykorzystamy zbiór danych, dotyczący pasażerów Titanica.

3.3 Wyniki

```
head(titanic.dane)
```

W tabeli 14, przedstawiającej przykładowe dane, możemy zobaczyć, że wśród zmiennych występują nie tylko cechy numeryczne, takie jak wiek(`Age`), ale również cechy kategoryczne, w tym:

- płeć(`Sex`), jako zmienna binarna,
- port(`Fare`), w którym pasażer wszedł na pokład statku, jako zmienna nominalna,
- klasę(`Pclass`), którą podróżował pasażer, jako zmienna uporządkowana.

Ponadto w danych istnieje zmienna grupująca `Survived`, która informuje o tym, czy pasażer przeżył katastrofę.

Do dalszej analizy będziemy wykorzystywać dane bez zmiennej grupującej, aby później na jej podstawie ocenić jakość odwzorowania.

```
titanic.dane.mds <- subset(titanic.dane, select=-c(Survived))
```

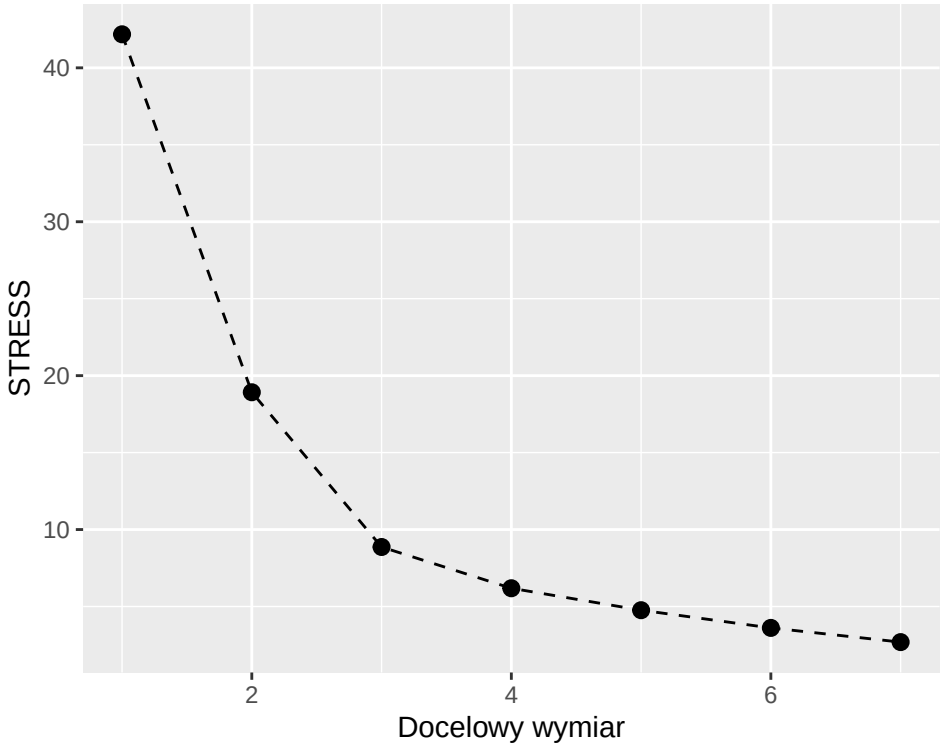
W kolejnych krokach, aby dokonać skalowania, będziemy potrzebować odmienności między wektorami cech.

```
macierz.odmiennosci <- as.matrix(daisy(
  titanic.dane.mds,
  stand=TRUE
))
```

Dysponując macierzą odmienności, możemy zbadać jak zmieniają się diagramy Shepparda oraz wartość funkcji kryterialnej `STRESS`, wraz ze zmianą wymiaru docelowej przestrzeni. Dlatego teraz wyznaczamy `STRESS` oraz diagram Shepparda dla k=1,2,. . . ,7.

```
badanie.MDS <- function (k.max, macierz.odmiennosci) {
  STRESS <- numeric(k.max)
  wykresy <- list()
  for (k in 1:k.max) {
    mds <- isoMDS(macierz.odmiennosci, k=k)
    STRESS[k] <- mds$stress
    odleglosci.k <- as.matrix(dist(mds$points, method="euclidean"))
    wykresy[[k]] <- ggplot() + geom_point(aes(x=c(macierz.odmiennosci), y=c(odleglosci.k
  }
  return(list(wykresy=wykresy, STRESS=STRESS))
}
k.max <- length(titanic.dane.mds)
wyniki <- badanie.MDS(k.max, macierz.odmiennosci)
```

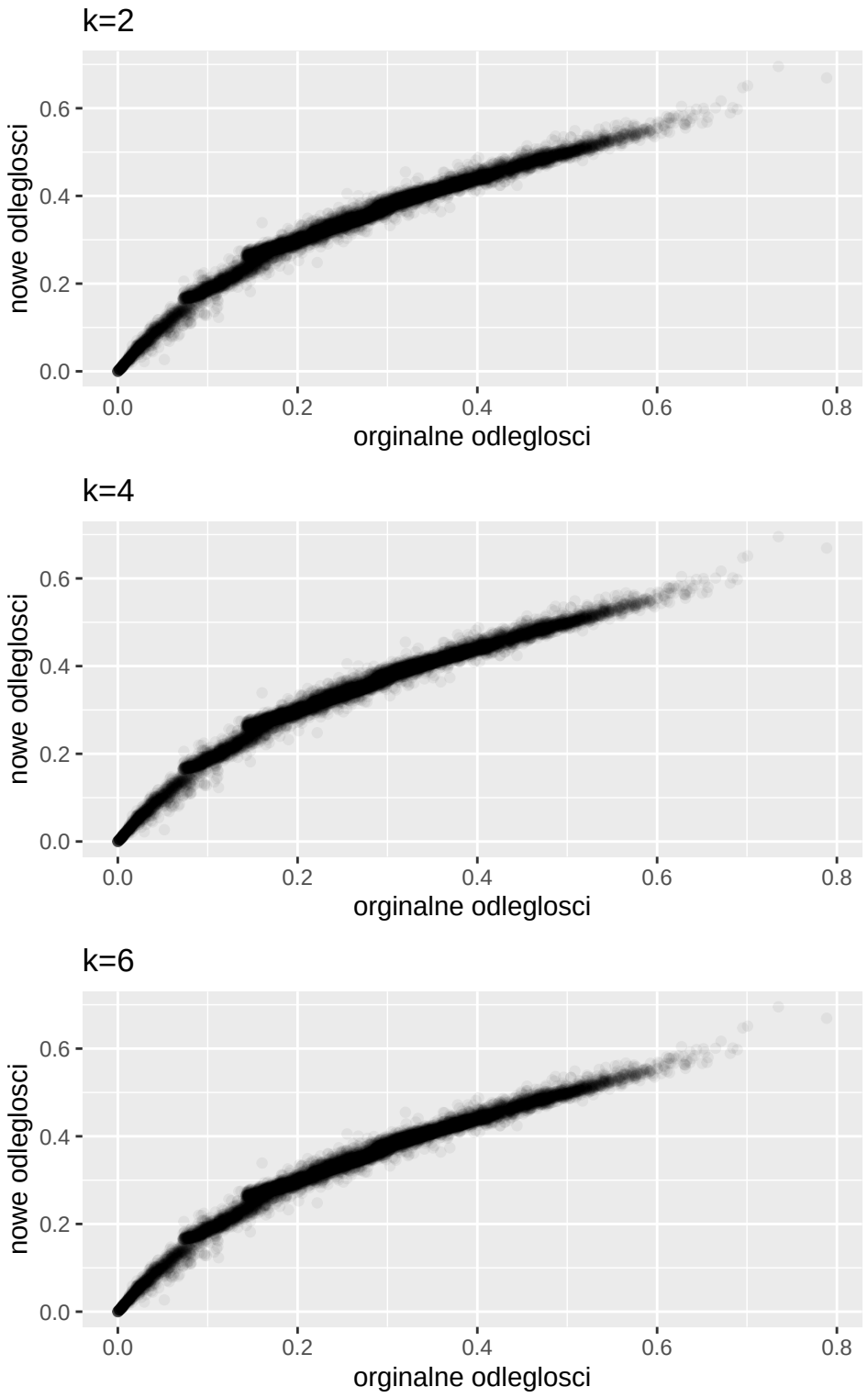
```
ggplot() + geom_line(aes(x=1:k.max, y=wyniki$STRESS))
```



Rysunek 17: Zależność funkcji kryterialnej STRESS od docelowego wymiaru

Na wykresie 17 widzimy, że funkcja `STRESS` maleje z każdym zwiększeniem wymiaru docelowego. Na jego podstawie, możemy również stwierdzić, że dla w przestrzeni dwuwymiarowej przyjmuje ona stosunkowo duże wartości, co wskazuje na potencjalnie dużą utratę informacji. Widzimy również, że także w przestrzeni trójwymiarowej występuje utrata informacji, jednak jest ona znacznie mniejsza, niż dla $k = 2$. Stąd, możemy przypuszczać, że dzięki wykorzystaniu metod wizualizacji 3D, będziemy mogli poznać strukturę zbioru danych znacznie lepiej, niż na standardowym, dwuwymiarowym wykresie.

```
wykresy <- wyniki$wykresy
ggarrange(plotlist=wykresy[c(2, 4, 6)], nrow=3)
```



Rysunek 18: Diagramy Shepparda

Diagramy Shepparda, które przedstawiają porównanie odległości między orginalną, a docelową przestrzeną, nie przedstawiają istotnych zmian między docelowymi wymiarami. Dlatego też, korzystając przede wszystkim z wykresu funkcji `STRESS`, jako docelowy wymiar przestrzeni, wybrałbym 4, ponieważ wykres zaczyna się w tym miejscu wypłaszczać, a jednocześnie pozwala on na wizualne badanie elementów zbioru poprzez wprowadzenie koloru lub kształtu, reprezentującego czwartą współrzędną w tej przestrzeni.

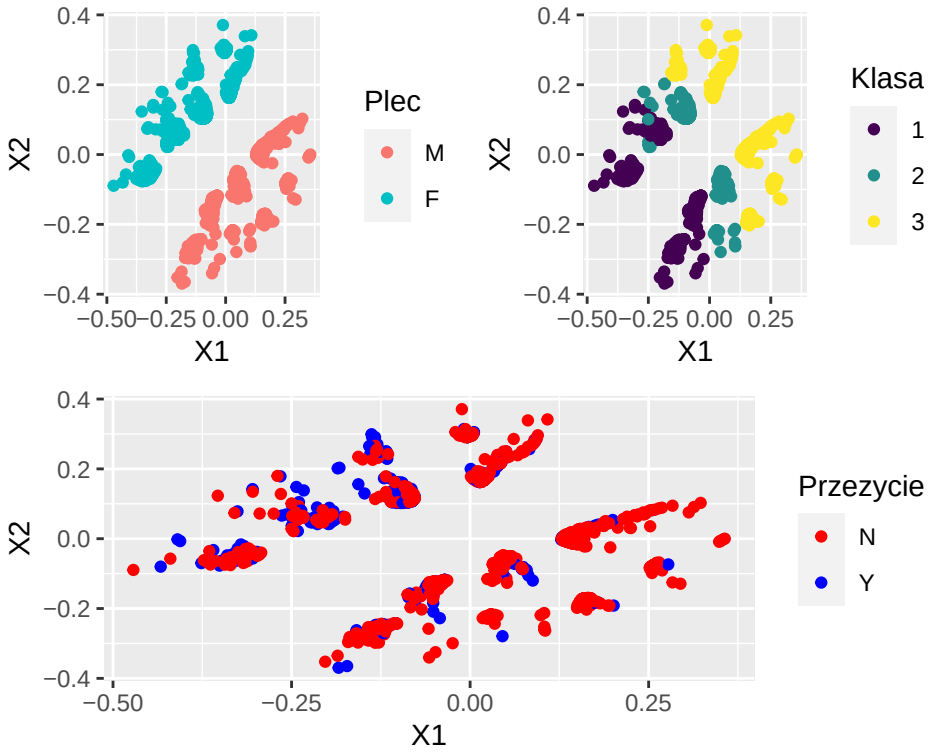
Tabela 15: Tabela kontygencji dla podziału płeć-przeżycie

	N	Y
M	553	100
F	204	213

Teraz spróbujemy ocenić jakość odwzorowań, korzystając z dodatkowej informacji, jaką jest zmienna grupująca `Survived`.

```
mds.k2 <- isoMDS(macierz.odmiennosci, k=2)
reprezentacja.mds.k2 <- data.frame(mds.k2$points)

ggplot(reprezentacja.mds.k2, aes(x=X1, y=X2)) +
  geom_point(aes(col=...))
```



Rysunek 19: Wykres po odwzorowaniu MDS dla $k = 2$ z podzialem na grupy

Widzimy, że po odwzorowaniu można wyróżnić dwie grupy, które rodzielają dane ze względu na płeć pasażera. Grupy te nie rozdzielają jednak danych ze względu na informację dotycząca tego, czy pasażer przeżył. Możemy jednak zauważyć, że istotnie wśród kobiet odsetek przeżycia jest dużo wyższy. Stąd, po sprawdzeniu, jak prezentuje się ich rozkład:

```
table(titanic.dane$Sex, titanic.dane$Survived)
```

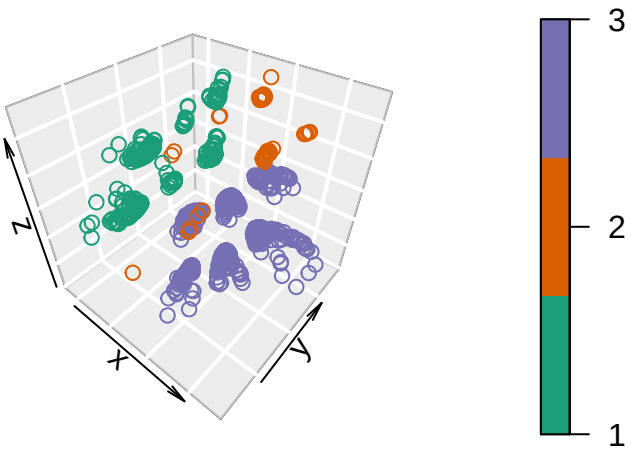
W tabeli 15 widzimy, że faktycznie, przypisując tym dwóm grupom odpowiednią klasę grupującą, otrzymalibyśmy skuteczność na poziomie 71.59%. Stąd możemy powiedzieć, że dane po MDS wciąż zachowują dość dużo informacji.

Ponadto, na wykresie 15 widzimy, że w danych pozostał dość dobry podział ze względu na klasę, którą podróżował pasażer.

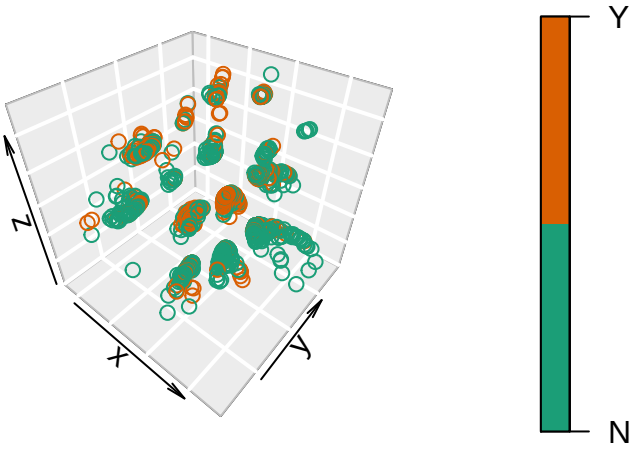
Teraz, możemy sprawdzić, czy przekształcenie MDS do przestrzeni trojwymiarowej będzie zawierało więcej informacji.

```
mds.k3 <- isoMDS(macierz.odmiennosci, k=3)
attach(data.frame(mds.k3$points))
```

```
points3D(X1,X2,X3,...)
```



Rysunek 20: Wykres po odwzorowaniu MDS dla $k = 3$ z podzialem na początek podróży



Rysunek 21: Wykres po odwzorowaniu MDS dla $k = 3$ z podziałem na przeżycie

Na wykresie 20 widzimy, że dodatkowy wymiar dostarcza nam informacji między innymi na temat portu, w którym pasażer wsiadł na statek.

Nie jest to jednak wystarczające do separacji pasażerów, którzy przeżyli, od tych, którzy nie przeżyli (Rys. 21). Mimo tego, redukcja wymiaru zdecydowanie pomogła w wizualizacji i w lepszym zrozumieniu danych.

3.4 Podsumowanie

Na podstawie przeprowadzonego eksperymentu, możemy stwierdzić, że skalowanie wielowymiarowe może być niezwykle pomocne podczas analizy danych. Mimo tego, że nie rozwiązało ono bezpośrednio głównego problemu, który wiąże się z tym zbiorem danych, czyli predykcji przeżycia pasażerów, to pozwoliło na przeanalizowanie danych w 2- i 3- wymiarowych przestrzeniach, co pozwoliło na sformułowanie początkowych hipotez, które mogłyby posłużyć do dalszej analizy.

4 Podsumowanie