Raport 2

Paweł Matławski
album 249732

13 marca 2021

Spis treści

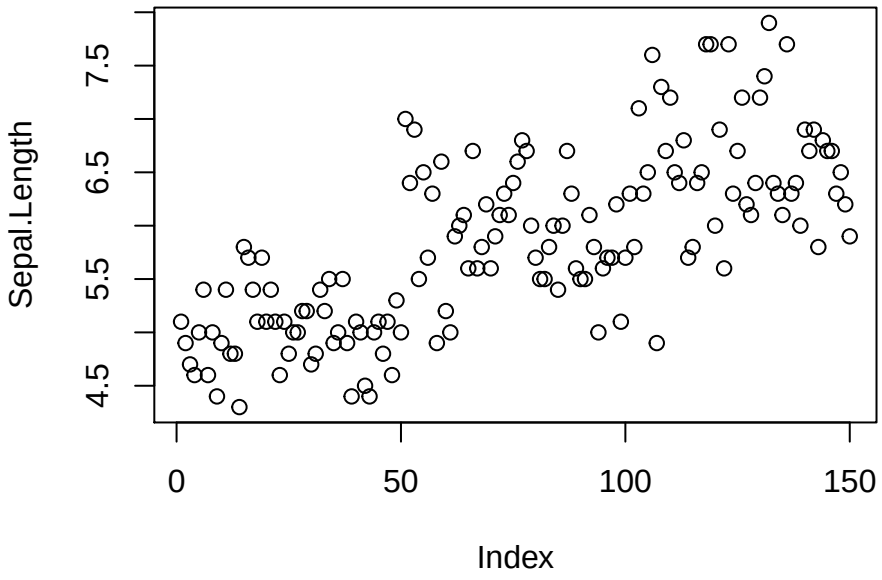
1 Dyskretyzacja (przedziałowanie) cech ciągłych

1.1 Dane

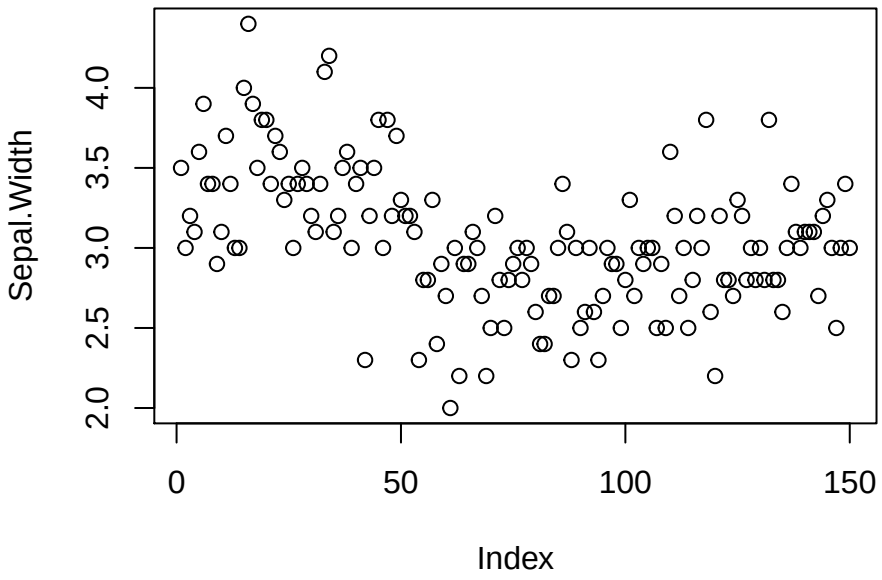
```
library('datasets')
attach(iris)
```

1.2 Wybór cech

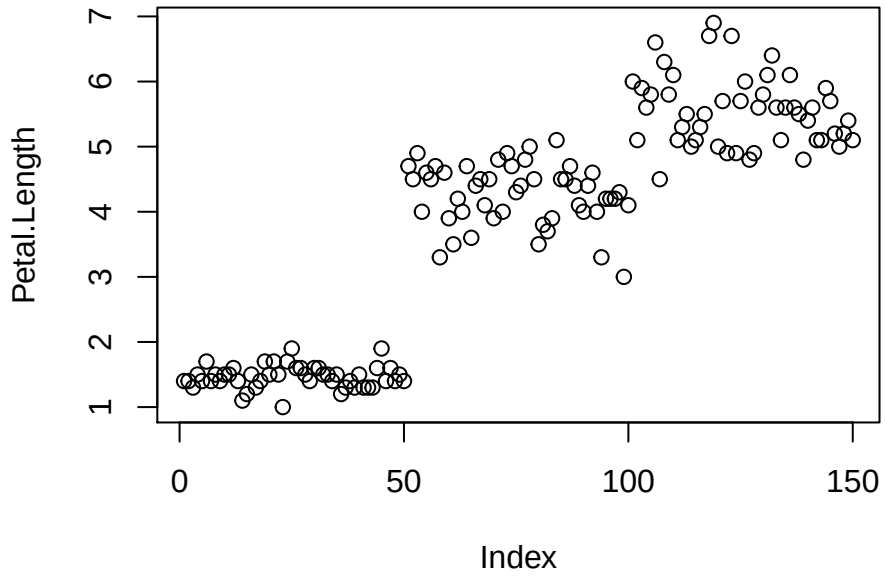
```
plot(Sepal.Length)
```



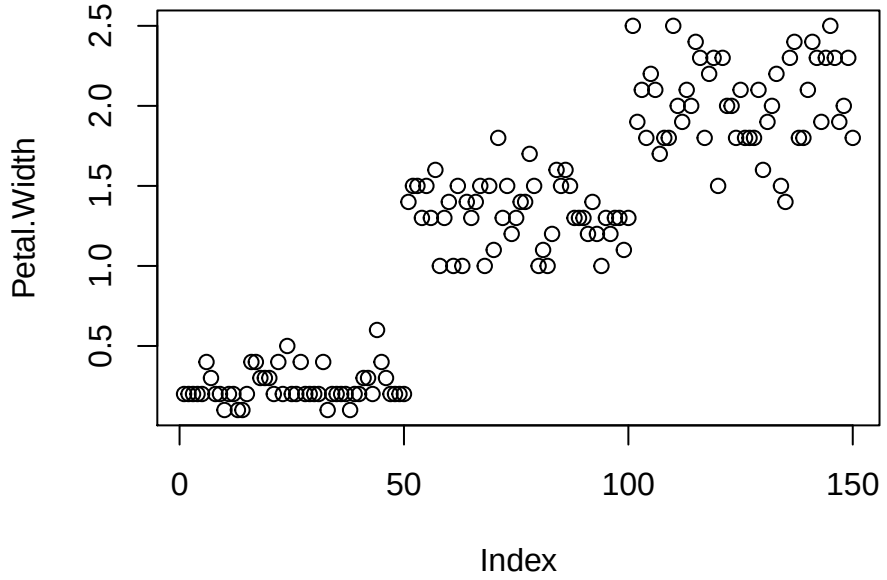

```
plot(Sepal.Width)
```



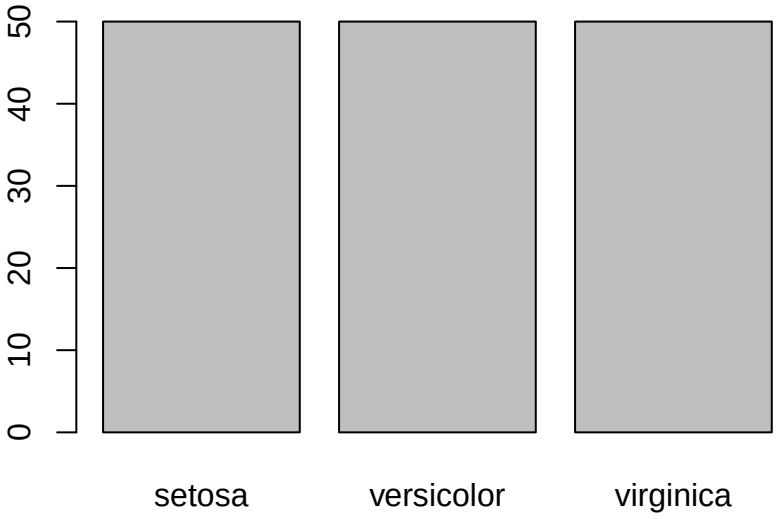

```
plot(Petal.Length)
```




```
plot(Petal.Width)
```




```
plot(Species)
```

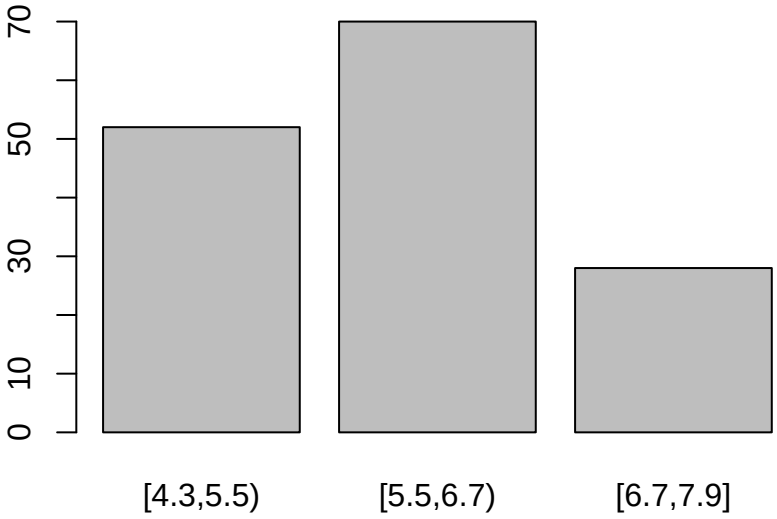


- Z powyższych wykresów możemy zauważyć, że cechy Petal są o wyższej zdolności dyskryminacyjnej, a cechy Sepal o niższej.

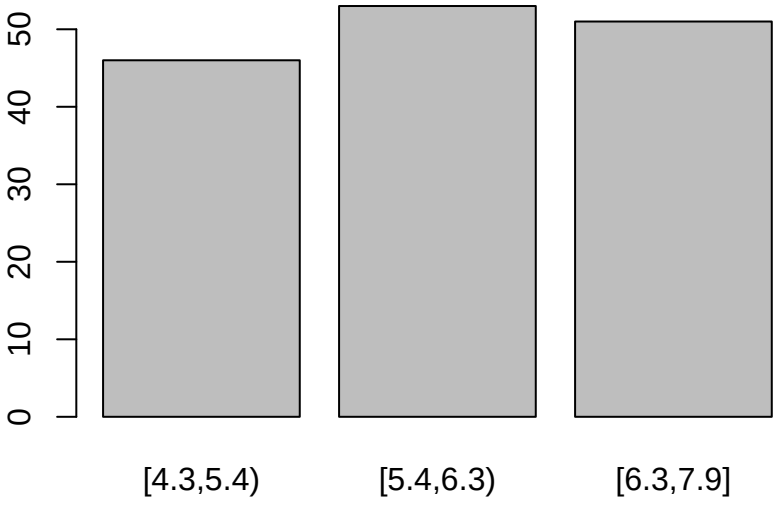
1.3 Porównanie nienadzorowanych metod dyskretyzacji

- Wybrałem cechy: jako lepszą Petal.Length, jako gorszą Sepal.Length. Zastosujemy teraz różne metody dyskretyzacji nienadzorowanej i porównamy metody.

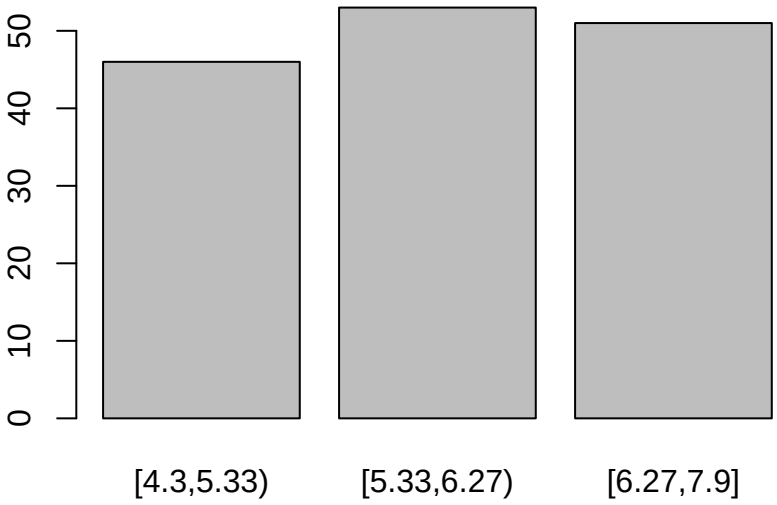
```
library('arules')

## Loading required package: Matrix
##
## Attaching package: 'arules'
## The following objects are masked from 'package:base':
##
##     abbreviate, write

d11 <- discretize(Sepal.Length, method = "interval", breaks = 3)
plot(d11)
```



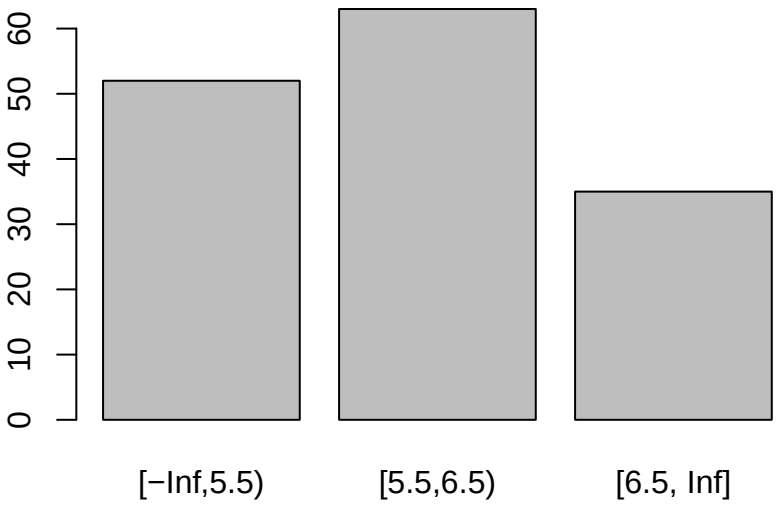

```
d12 <- discretize(Sepal.Length, method = "frequency", breaks = 3)
plot(d12)
```



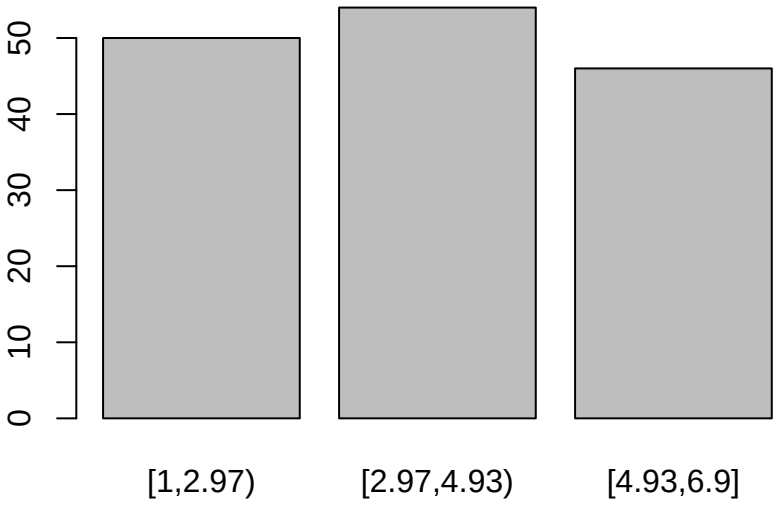

```
d13 <- discretize(Sepal.Length, method = "cluster", breaks = 3)
plot(d13)
```



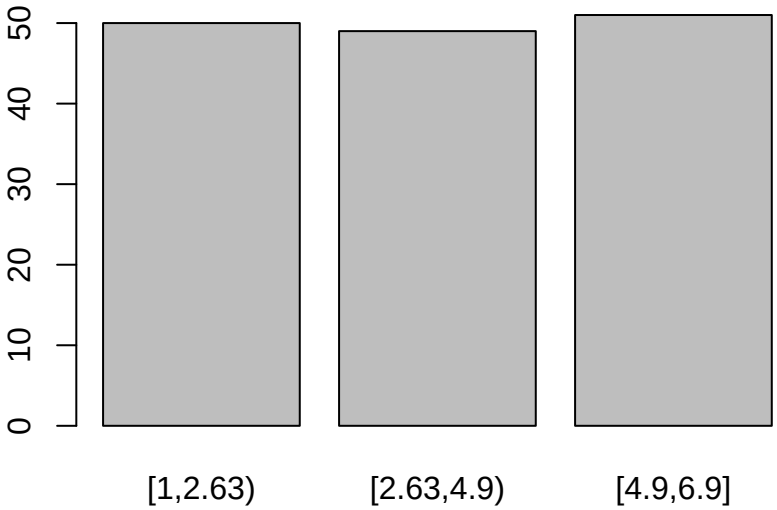

```
d14 <- discretize(Sepal.Length, method = "fixed", breaks = c(-Inf, 5.5 , 6.5, Inf))
plot(d14)
```



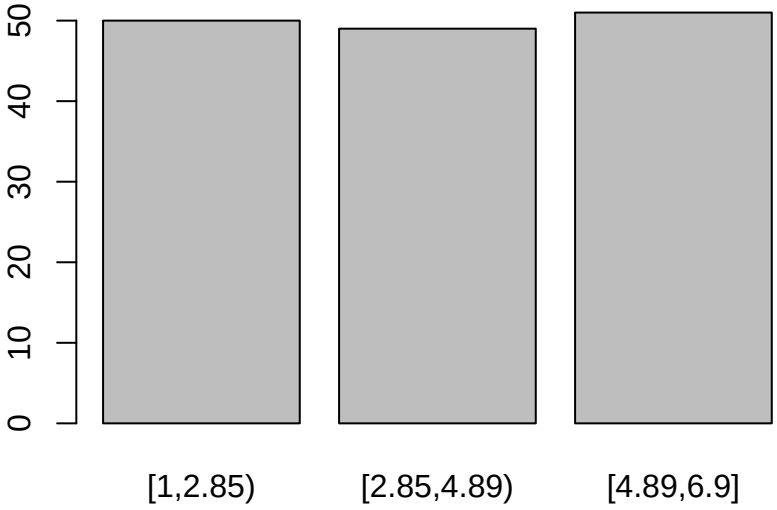

```
d31 <- discretize(Petal.Length, method = "interval", breaks = 3)
plot(d31)
```



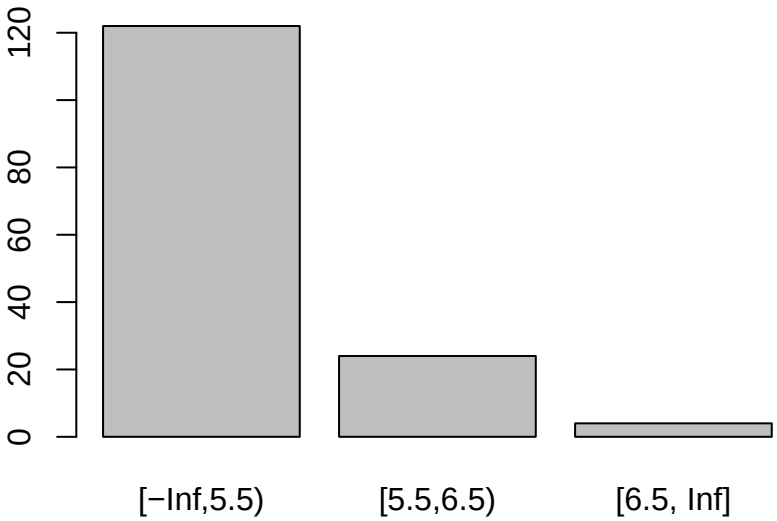

```
d32 <- discretize(Petal.Length, method = "frequency", breaks = 3)
plot(d32)
```




```
d33 <- discretize(Petal.Length, method = "cluster", breaks = 3)
plot(d33)
```




```
d34<- discretize(Petal.Length, method = "fixed", breaks = c(-Inf, 5.5 , 6.5, Inf))
plot(d34)
```

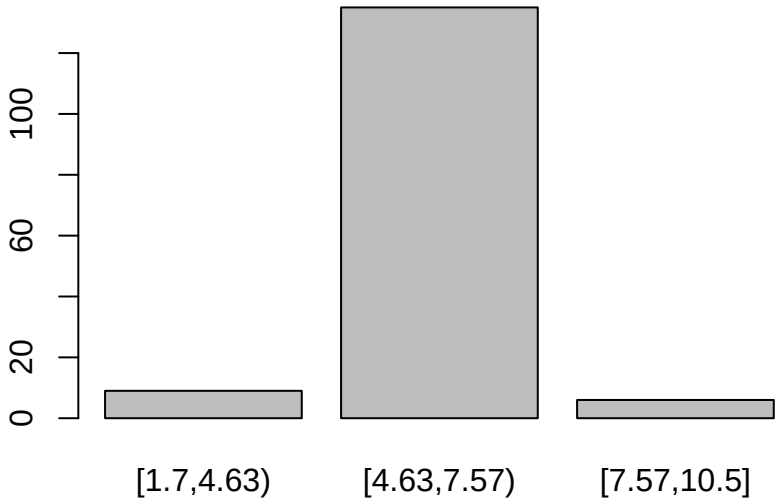


- Analizując powyższe rezultaty, możemy zauważyć że najskuteczniejszymi metodami są metody ”k-means” i ”equal frequency”

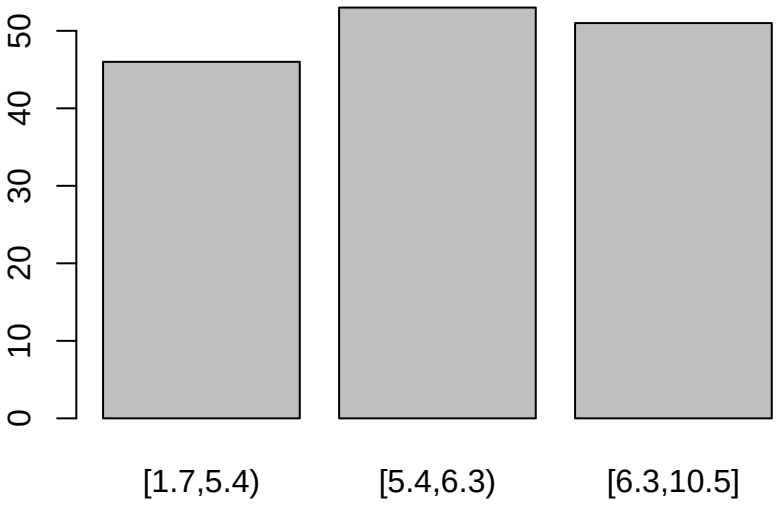
1.4 Wpływ obserwacji odstających

- Zastępując wartości największe i najmniejsze wartościami odstającymi, zbadamy ich wpływ na wyniki dyskretyzacji.

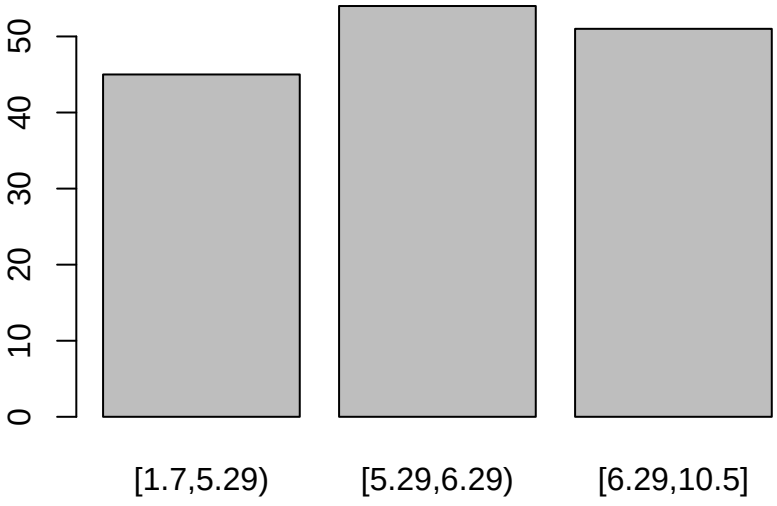
```
iris2 <- data.frame(iris)
iris2$Sepal.Length[which.min(iris2$Sepal.Length)] <- min(iris2$Sepal.Length) - 2*IQR(iri
iris2$Sepal.Length[which.max(iris2$Sepal.Length)] <- max(iris2$Sepal.Length) + 2*IQR(iri
e11 <- discretize(iris2$Sepal.Length, method = "interval", breaks = 3)
plot(e11)
```



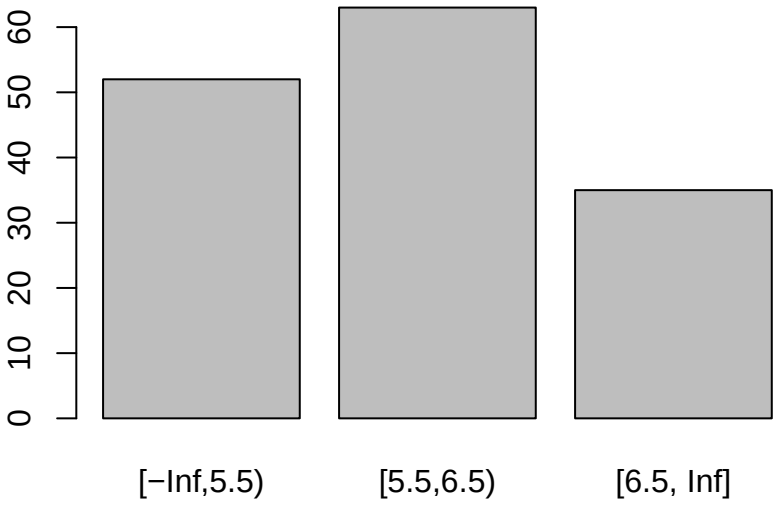

```
e12 <- discretize(iris2$Sepal.Length, method = "frequency", breaks = 3)
plot(e12)
```



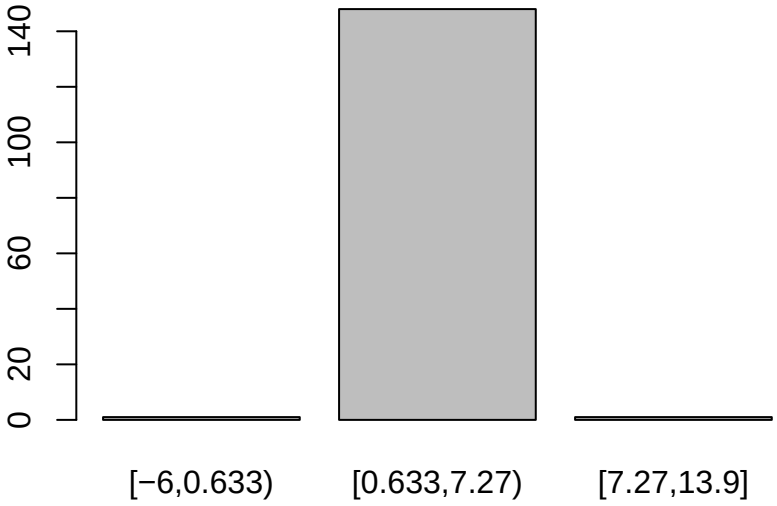

```
e13 <- discretize(iris2$Sepal.Length, method = "cluster", breaks = 3)
plot(e13)
```



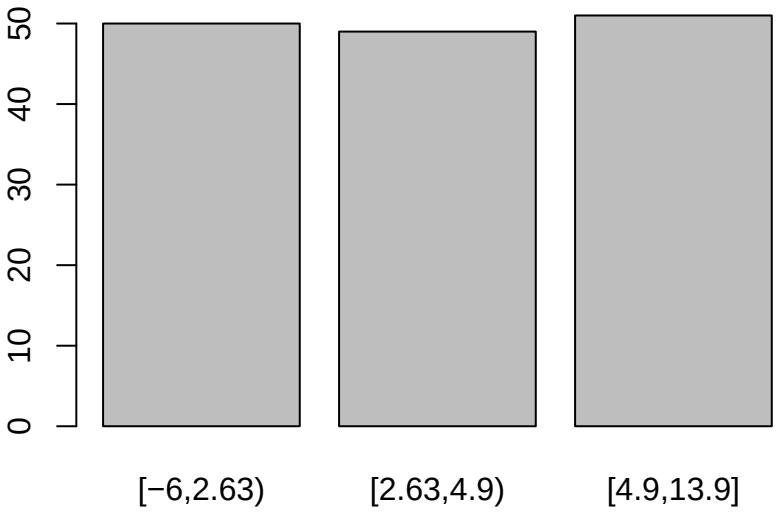

```
e14 <- discretize(iris2$Sepal.Length, method = "fixed", breaks = c(-Inf, 5.5 , 6.5, Inf))
plot(e14)
```



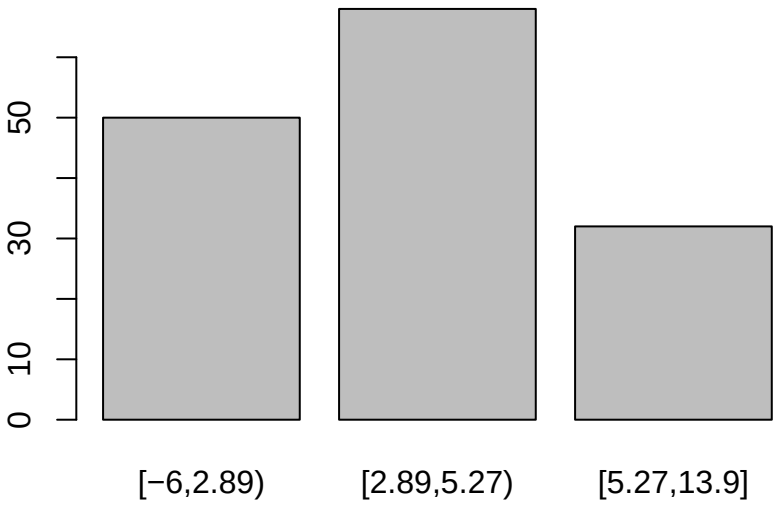

```r
iris2$Petal.Length[which.min(iris2$Petal.Length)] <- min(iris2$Petal.Length) - 2*IQR(iri
iris2$Petal.Length[which.max(iris2$Petal.Length)] <- max(iris2$Petal.Length) + 2*IQR(iri
e31 <- discretize(iris2$Petal.Length, method = "interval", breaks = 3)
plot(e31)
```



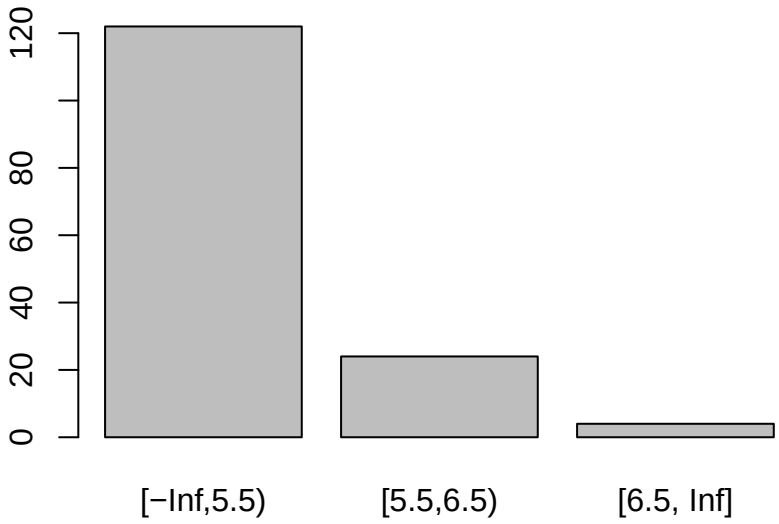
```
e32 <- discretize(iris2$Petal.Length, method = "frequency", breaks = 3)
plot(e32)
```




```
e33 <- discretize(iris2$Petal.Length, method = "cluster", breaks = 3)
plot(e33)
```



```
e34 <- discretize(iris2$Petal.Length, method = "fixed", breaks = c(-Inf, 5.5 , 6.5, Inf))
plot(e34)
```



- Zauważamy, że wartości odstające negatywnie wpływają na wyniki dyskretyzacji dla wszystkich metod, poza metodą "equal frequecy".

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

2.1 Dane

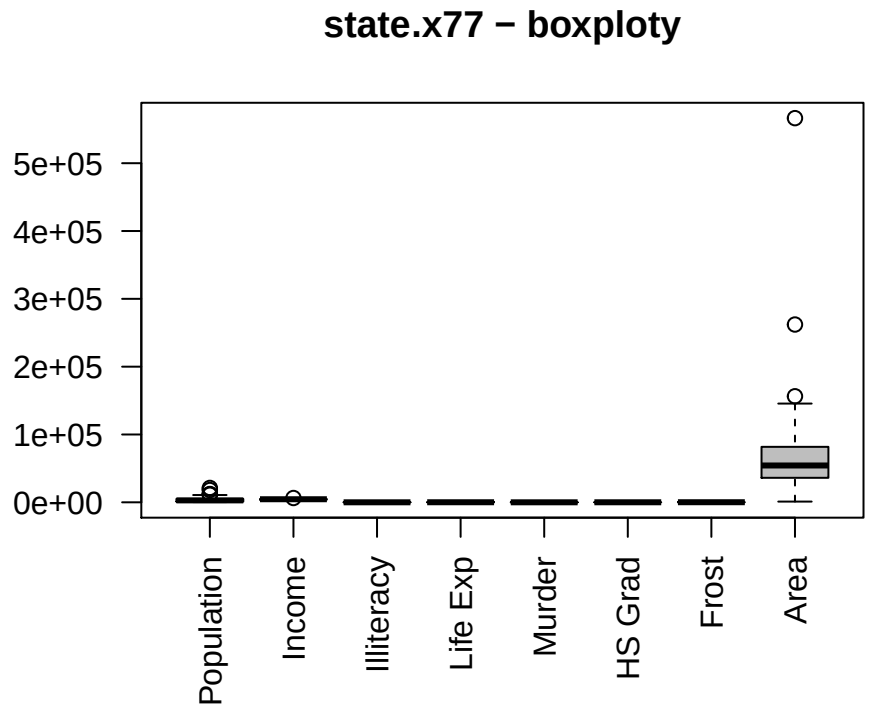
- Będziemy pracować na zbiorze danych ¡i¿state.x77¡/i¿, który zawiera podstawowe informacje o 50 stanach USA.

```
library('datasets')
dane<-as.data.frame(state.x77)
```

2.2 Przygotowanie danych

- Za pomocą wykresów pudełkowych porównamy zmienność poszczególnych cech.

```r
boxplot(dane$Population, dane$Income, dane$Illiteracy,
        dane$`Life Exp`, dane$Murder, dane$`HS Grad`,
        dane$Frost, dane$Area,
        col = 1:8,
        main = "state.x77 - boxploty",
        names = c("Population","Income","Illiteracy","Life Exp",
                  "Murder","HS Grad","Frost","Area"),
        las = 2)
```



- Wyraźnie zauważamy, że zarówno wariancja, jak i mediana zmiennej Area wyraźnie różni się od pozostałych, więc niezbędne będzie zastosowanie standaryzacji.

2.3 Wyznaczenie składowych głównych

```r
dane.pca <- prcomp(state.x77, scale. = T, center = T, retx = T)
```

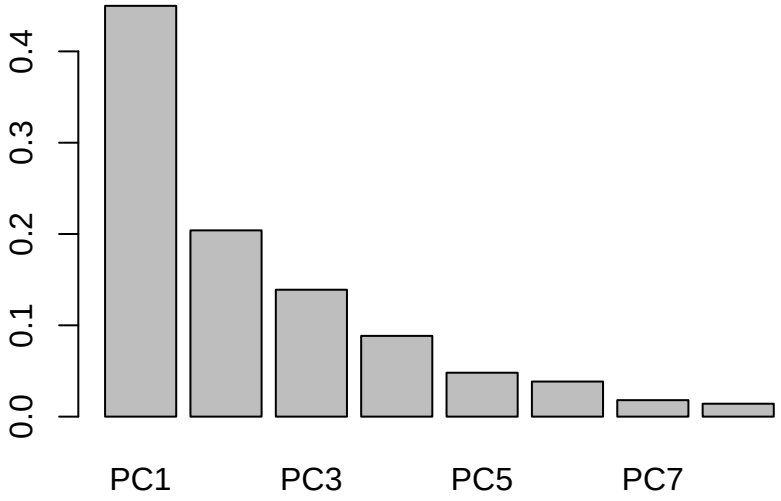
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Population	0.13	0.41	-0.66	-0.41	0.41	-0.01	-0.06	-0.22
Income	-0.30	0.52	-0.10	-0.09	-0.64	0.46	0.01	0.06
Illiteracy	0.47	0.05	0.07	0.35	0.00	0.39	-0.62	-0.34
Life Exp	-0.41	-0.08	-0.36	0.44	0.33	0.22	-0.26	0.53
Murder	0.44	0.31	0.11	-0.17	-0.13	-0.33	-0.30	0.68
HS Grad	-0.42	0.30	0.05	0.23	-0.10	-0.64	-0.39	-0.31
Frost	-0.36	-0.15	0.39	-0.62	0.22	0.21	-0.47	0.03
Area	-0.03	0.59	0.51	0.20	0.50	0.15	0.29	0.01

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Standard deviation	1.8971	1.2775	1.0545	0.8411	0.6202	0.5545	0.3801	0.3364
Proportion of Variance	0.4499	0.2040	0.1390	0.0884	0.0481	0.0384	0.0181	0.0141
Cumulative Proportion	0.4499	0.6539	0.7928	0.8813	0.9294	0.9678	0.9859	1.0000

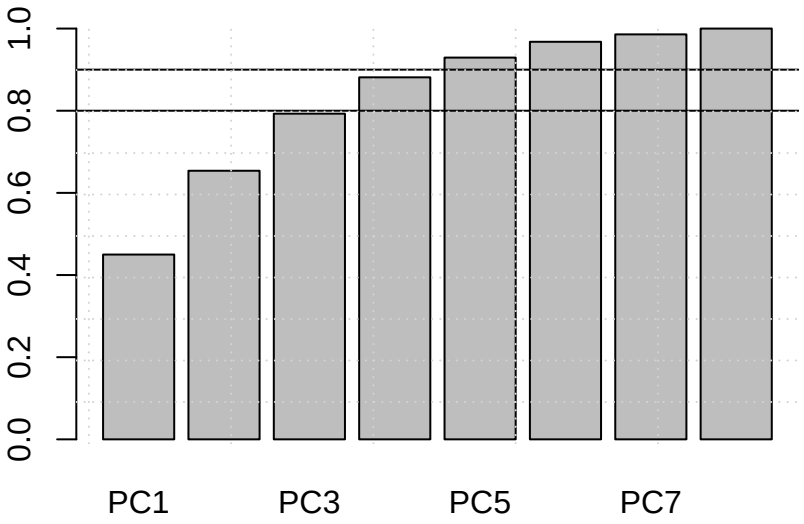
- 1-szy wektor ładunków przypisuje w przybliżeniu jednakową wagę zmiennym Illiteracy, Murder, HS Grad i Life Exp i najmniejszą wagę zmiennej Area. Drugi największą wagę przypisuje zmiennym Area oraz Income, a najmniejszą Illiteracy i Life Exp. Trzeci zaś związany jest głównie ze zmiennymi Population i Area, a najmniejszą wagę przypisuje zmiennym HS Grad oraz Illiteracy.

2.4 Zmienność odpowiadająca poszczególnym składowym

```
wariancja <- (dane.pca$sdev ^2)/sum(dane.pca$sdev^2)
laczna.wariancja <- cumsum(wariancja)
barplot(wariancja, names.arg = c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8"))
```



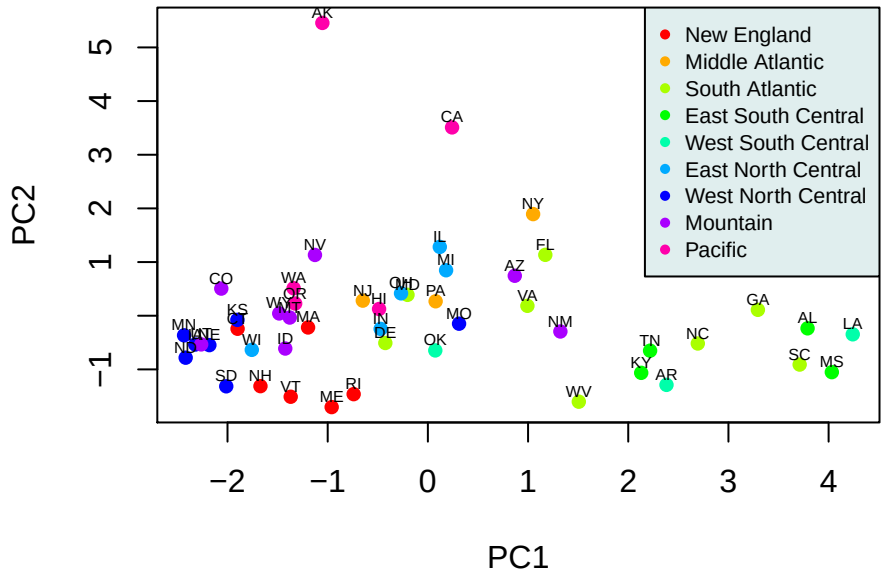

```
barplot(laczna.wariancja, names.arg = c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8"))
abline(h=0.8)
abline(h=0.9)
grid(ny = 10)
```



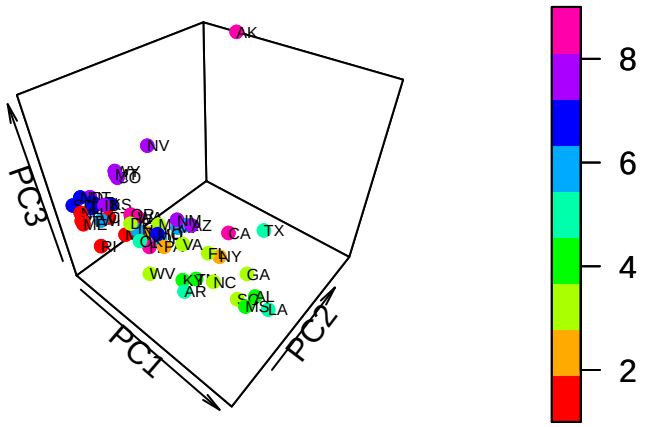
- PC1 wyjaśnia 45 procent całkowitej zmienności danych, a PC2 – 20 procent.
- Do wyjaśnienia 80 procent zmienności danych potrzebujemy co najmniej 4 składowych głównych, a do wyjaśnienia 90 procent 5 składowych.

2.5 Wizualizacja danych wielowymiarowych

```
kolory <- rainbow(9)

plot(dane.pca$x[,1], dane.pca$x[,2], col=kolory[as.numeric(state.division)], pch=16, xla
text(dane.pca$x[,1], dane.pca$x[,2]+0.2, labels=state.abb, cex=0.5)
legend("topright",legend=levels(state.division), col=kolory, pch=16, cex=0.7, bg="azure2
```



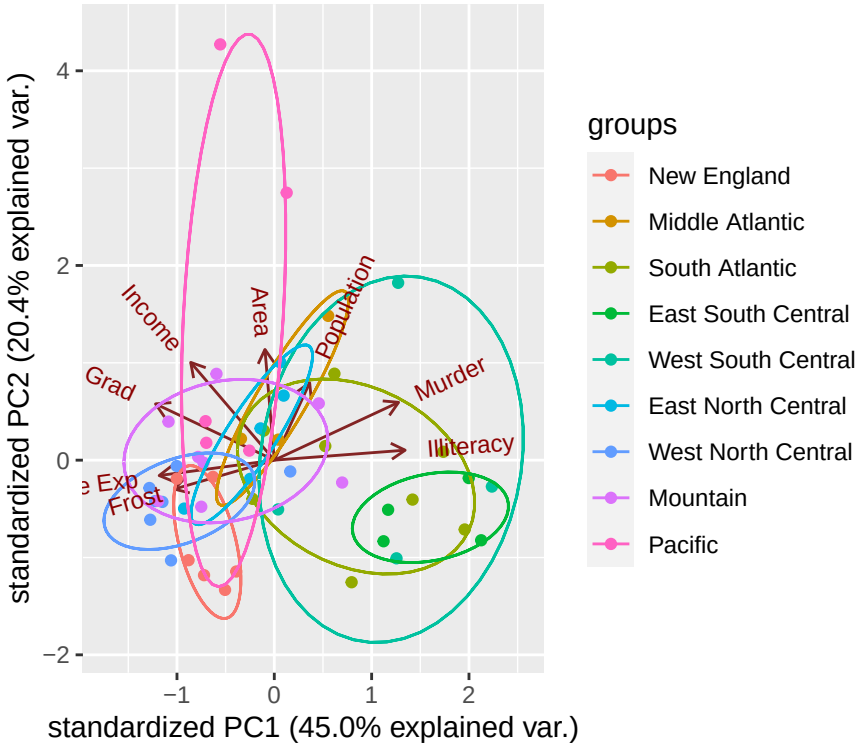

```
library(plot3D)
scatter3D(dane.pca$x[,1], dane.pca$x[,2], dane.pca$x[,3],  colvar=as.numeric(state.divis
text3D(dane.pca$x[,1], dane.pca$x[,2], dane.pca$x[,3],  labels = state.abb, add=TRUE, ce
```



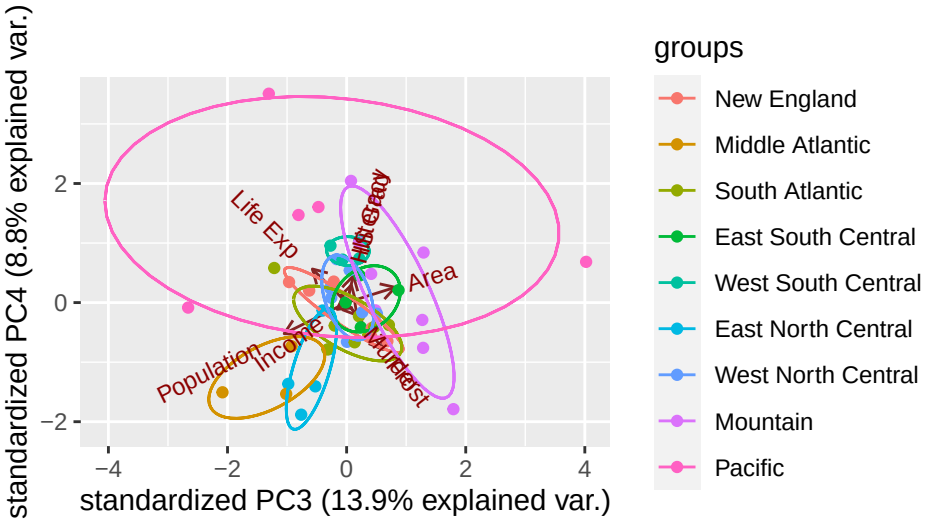

```
library(ggbiplot)

## Loading required package: ggplot2
```

```
## Loading required package: plyr
## Loading required package: scales
## Loading required package: grid
```

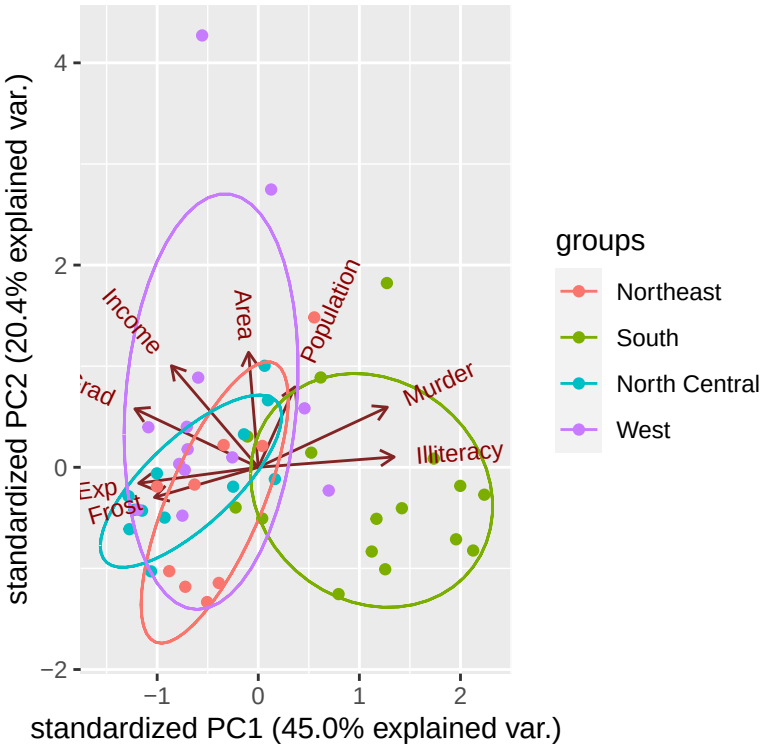
```
ggbiplot(dane.pca,ellipse=TRUE,choices=c(1,2), groups=state.division)
```



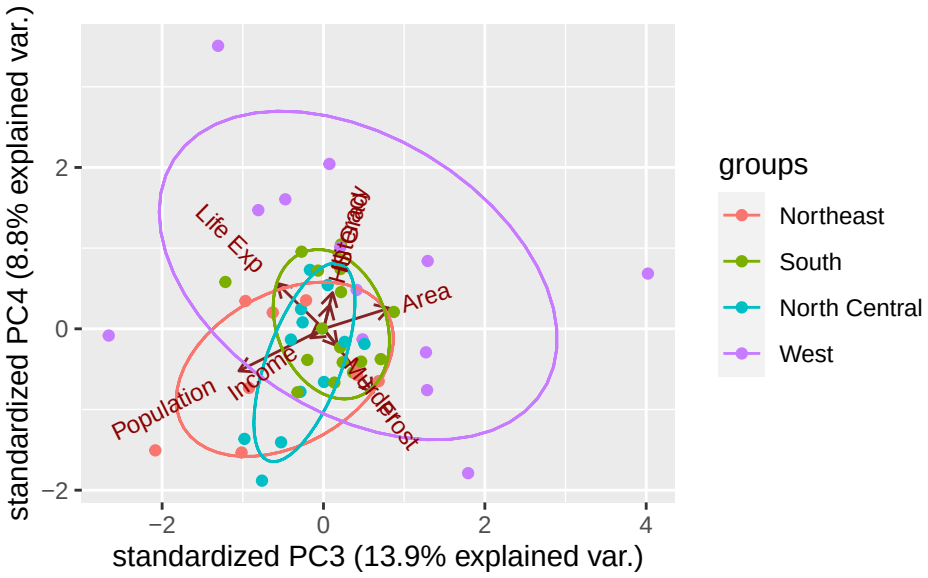

```
ggbiplot(dane.pca,ellipse=TRUE,choices=c(3,4), groups=state.division)
```



```
ggbiplot(dane.pca,ellipse=TRUE,choices=c(1,2), groups=state.region)
```




```
ggbiplot(dane.pca,ellipse=TRUE,choices=c(3,4), groups=state.region)
```



- Wykres rozrzutu w przestrzeni pokazuje nam, że najbardziej wyróżniającymi się stanami są Alaska, Kalifornia, Teksas oraz Luizjana. Trzy pierwsze z wymienionych stanów

są zdecydowanie największymi pod względem powierzchni. Alaska wyróżnia się również niewielką liczbą mieszkańców. Luizjana zaś cechuje się zaś najwyższym stopniem analfabetyzmu, a także sporą liczbą morderstw i niewielkim dochodem na jednego mieszkańca.

- Stany na zachodzie kraju są bardzo zróżnicowane pod względem powierzchni. Te na południu wyróżniają się wysokim stopniem analfabetyzmu i sporą liczbą popełnianych morderstw.

2.6 Korelacja zmiennych

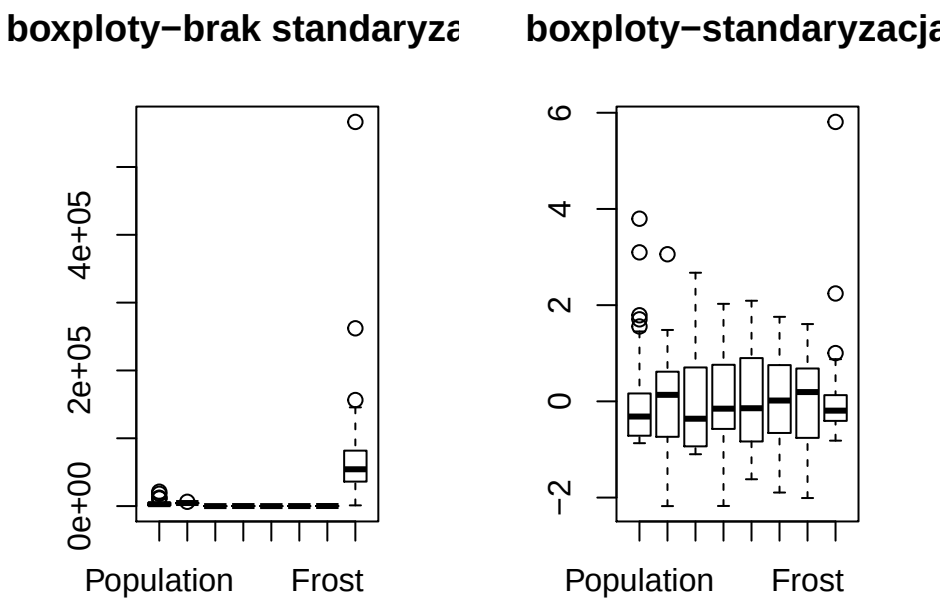
```
cor <- cor(dane)
```

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
Population	1.00	0.21	0.11	-0.07	0.34	-0.10	-0.33	0.02
Income	0.21	1.00	-0.44	0.34	-0.23	0.62	0.23	0.36
Illiteracy	0.11	-0.44	1.00	-0.59	0.70	-0.66	-0.67	0.08
Life Exp	-0.07	0.34	-0.59	1.00	-0.78	0.58	0.26	-0.11
Murder	0.34	-0.23	0.70	-0.78	1.00	-0.49	-0.54	0.23
HS Grad	-0.10	0.62	-0.66	0.58	-0.49	1.00	0.37	0.33
Frost	-0.33	0.23	-0.67	0.26	-0.54	0.37	1.00	0.06
Area	0.02	0.36	0.08	-0.11	0.23	0.33	0.06	1.00

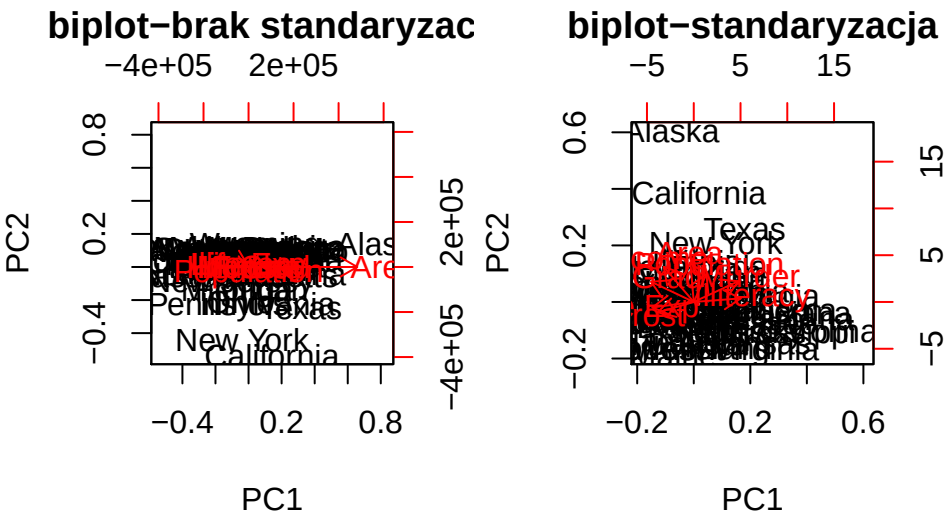
- Z wykresów oraz tabeli odczytujemy, że najsilniejsze korelacje występują pomiędzy średnią długością życia a liczbą popełnianych zabójstw. Zauważalne są również powiązania między stopniem analfabetyzmu a liczbą morderstw i procentem osób, które ukończyły szkołę średnią, a także, co ciekawe, średnią liczbą dni z temperaturą minimalną poniżej 0 stopni Celsjusza. Na średni dochód ma za to głóWnie wpływ procent osób, które ukończyły szkołę średnią.

2.7 Końcowe wnioski

```
dane.stand <- scale(dane)
par(mfrow=c(1,2))
boxplot(dane, main="boxploty-brak standaryzacji")
boxplot(dane.stand, main="boxploty-standaryzacja")
```



```
par(mfrow=c(1,2))
biplot(prcomp(state.x77, scale. = F), main="biplot-brak standaryzacji")

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

biplot(dane.pca, main="biplot-standaryzacja")
```



- Jak widzimy po powyższych wykresach standaryzacja miała bardzo duże znaczenie, ponieważ pierwotne wykresy są zdominowane przez zmienną Area, która ma wyraźnie większą wariancję od pozostałych.
- Stany Zjednoczone są bardzo zróżnicowanym krajem. Poszczególne zmienne potrafią się bardziej różnić w zależności od położenia danego stanu.
- Do otrzymania zadowalającej reprezentacji danych potrzebujemy minimum czterech składowych.

3 Skalowanie wielowymiarowe (Multidimensional Scaling (MDS))

3.1 Dane

- Posługując się znalezioną funkcją, znalazłem dataset który odpowiadałby wymogom zadania.

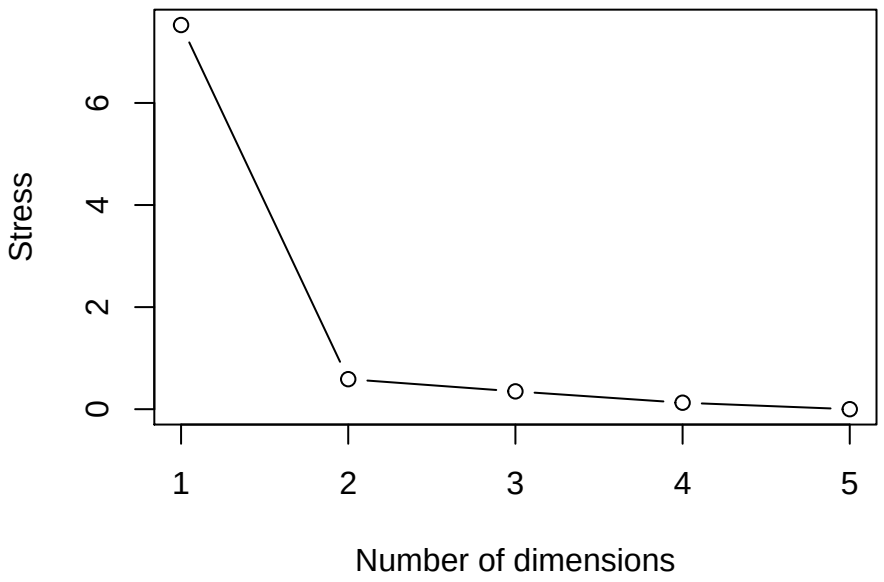
```
available_datasets <- data(package='datasets')[['results']][, 3];

meets_reqs <- function(datasets) {
sapply(datasets, function(ds) {
  dat <- get(sub(" .*", "", ds))
  hasfactors <- "factor" %in% sapply(dat, class)
  hasfactors
})
```

```
}

res <- meets_reqs(available_datasets)
res[res]

##                     CO2            ChickWeight           InsectSprays
##                    TRUE                   TRUE                   TRUE
##          OrchardSprays            PlantGrowth              Puromycin
##                    TRUE                   TRUE                   TRUE
##            ToothGrowth                 attenu                chickwts
##                    TRUE                   TRUE                   TRUE
##                  infert                   iris                    npk
##                    TRUE                   TRUE                   TRUE
##                   sleep state.division (state)   state.region (state)
##                    TRUE                   TRUE                   TRUE
##             warpbreaks
##                    TRUE

data(npk)
View(npk)
factorvar <- sapply(npk, is.factor)
factorvar

## block     N     P     K yield
##  TRUE  TRUE  TRUE  TRUE FALSE
```

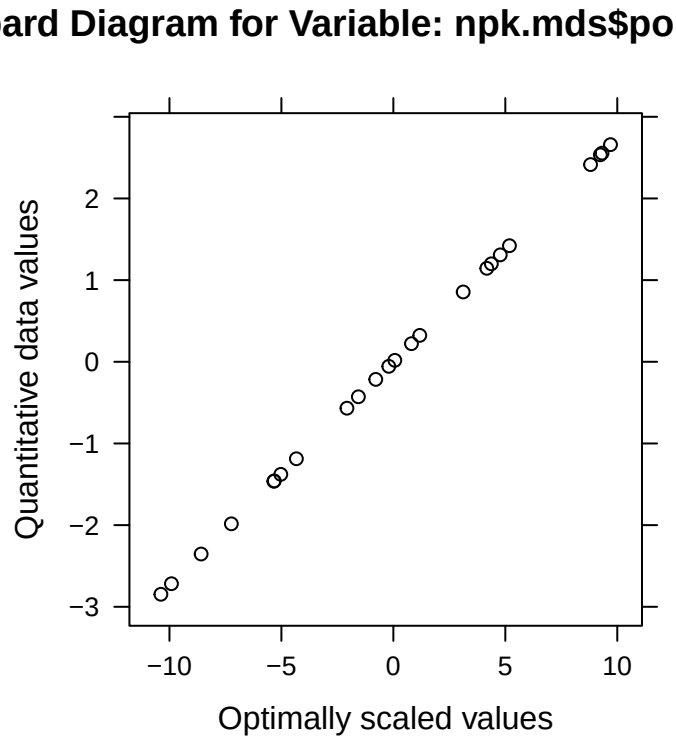
3.2 Redukcja wymiaru na bazie MDS

```
library("MASS")
library("optiscale")

## Loading required package: lattice

npk.dist <- dist(npk)
npk.mds <- isoMDS(npk.dist, k=2)

## initial  value 0.712301
## iter   5 value 0.590791
## final  value 0.588167
## converged

scree.plot = function(d, k) {
  stresses=isoMDS(d, k=k)$stress
  for(i in rev(seq(k-1)))
    stresses=append(stresses,isoMDS(d, k=i)$stress)
  plot(seq(k),rev(stresses), type="b", xaxp=c(1,k, k-1), ylab="Stress", xlab="Number of
}
scree.plot(npk.dist, k=5)
```

```
## initial  value 0.000000
## final  value 0.000000
## converged
## initial  value 0.400718
## iter   5 value 0.214257
## iter  10 value 0.155711
## iter  15 value 0.141583
## iter  20 value 0.130529
## final  value 0.126861
## converged
## initial  value 0.607137
## iter   5 value 0.401897
## iter  10 value 0.361144
## iter  15 value 0.353391
## iter  20 value 0.349573
## final  value 0.348928
## converged
## initial  value 0.712301
## iter   5 value 0.590791
## final  value 0.588167
## converged
## initial  value 8.068614
## final  value 7.526833
## converged
```



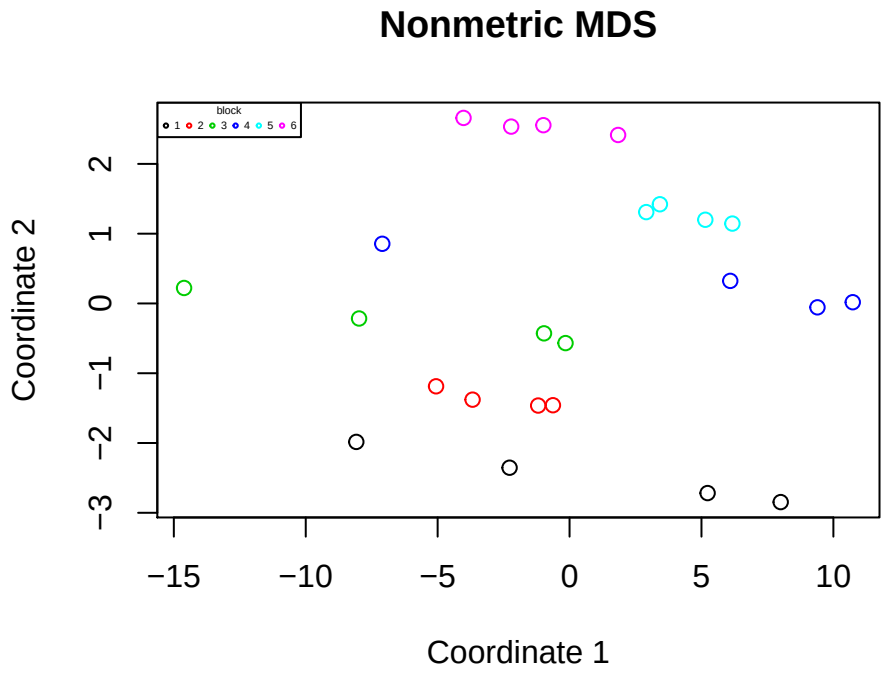
```
npk.mdsscaled <- opscale(x.qual =npk.mds$points[,1], x.quant = npk.mds$points[,2])
shepard(npk.mdsscaled, main.title= "Shepard Diagram")
```



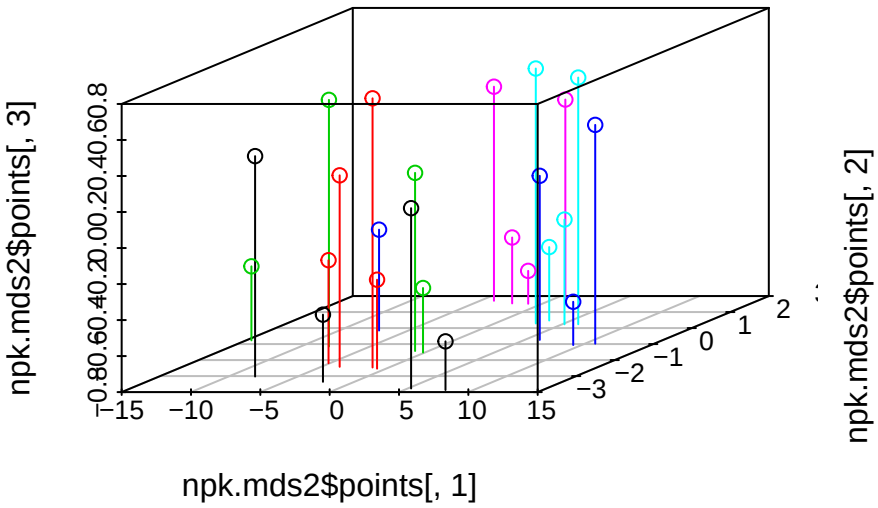
- Poprzez analizę funkcji STRESS zależnej od wymiaru oraz diagramu Sheparda widzimy, że wymiar d=2 jest optymalny dla naszego dataset.

3.3 Wizualizacja danych

```
library("scatterplot3d")
x <- npk.mds$points[,1]
y <- npk.mds$points[,2]
plot(x, y, xlab="Coordinate 1", ylab="Coordinate 2",
     main="Nonmetric MDS", col=npk$block)
legend("topleft", y=NULL, unique(npk$block),pch=1, horiz = TRUE,col=1:length(npk$block),
```




```
npk.mds2 <- isoMDS(npk.dist, k=3)

## initial  value 0.607137
## iter   5 value 0.401897
## iter  10 value 0.361144
## iter  15 value 0.353391
## iter  20 value 0.349573
## final  value 0.348928
## converged

scatterplot3d(npk.mds2$points[,1], npk.mds2$points[,2], z= npk.mds2$points[,3], color=np
```



- Możemy zauważyć, że cechy z bloku 1, ,3 i 4 wyróżniają się większą odmiennością, a więc świadczy to o mniejszej regularności wyników na temat wzrostu groszku (temat badań).