Eksploracjaa Danych raport 2

Karol Pustelnik
index 249828

27 kwietnia 2020

Spis treści

1 Krótki opis zagadnienia

W raporcie omówie metody redukcji wymiaru (PCA i MDS) na przykladzie dostepnego w R zbioru danych iris i state.x77

2 Opis eksperymentów i analiz

Uzyje przede wszystkim funkcji do redukcji wymiaru dla PCA i MDS. Oprócz tego uzyje:

- wykresów 3d,
- macierzy korelacji,
- dyskretyzacji zmiennych,

3 Wyniki

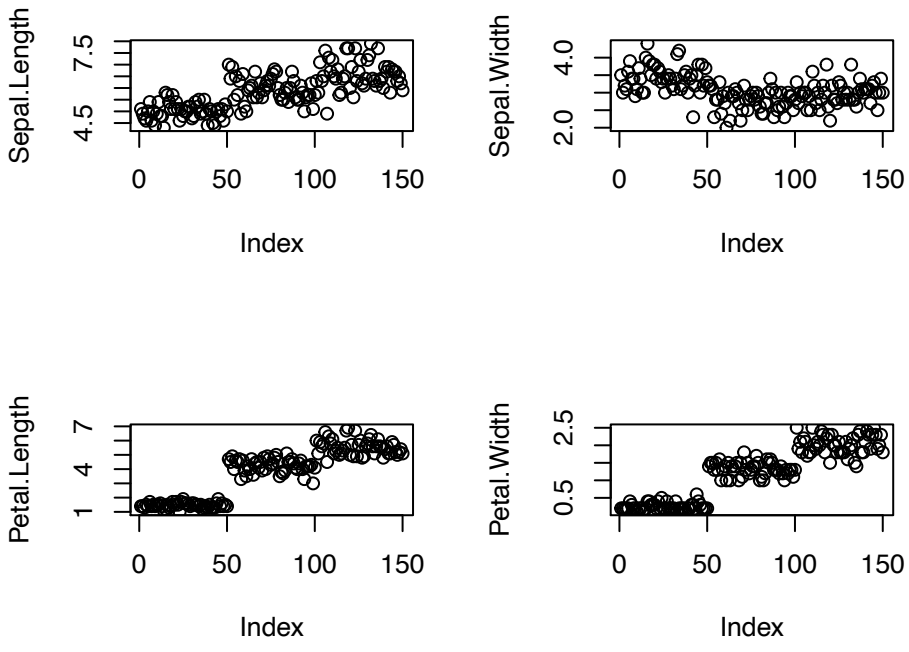
Zimportujmy najpierw biblioteki, z których bede korzystal i zaladujmy dane.

```
library("datasets")
library("arules")
library("NLP")
library("ggplot2")
library("dplyr")
library("plot3D")
library("corrplot")
library("e1071")
library("MASS")
library("cluster")
```

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

Narysujmy wykresy zmiennych

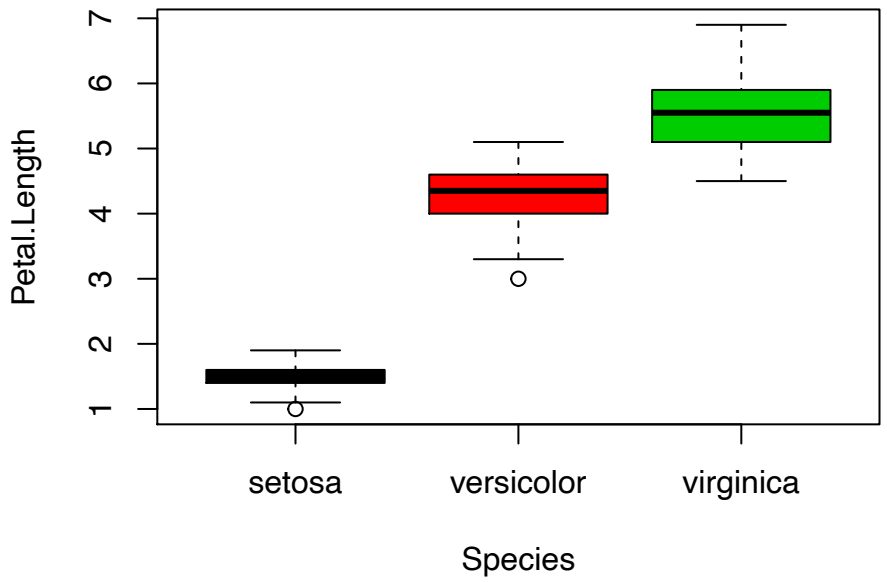
```
par(mfrow=c(2,2))
plot(Sepal.Length)
plot(Sepal.Width)
plot(Petal.Length)
plot(Petal.Width)
```



Zdolnosci dyskryminacyjne zmiennych w duzej mierze zaleza od tego czy te zmienne posiadaja rozklad przyblizony do normalnego. Jak widac z wykresów, zmienna której rozklad

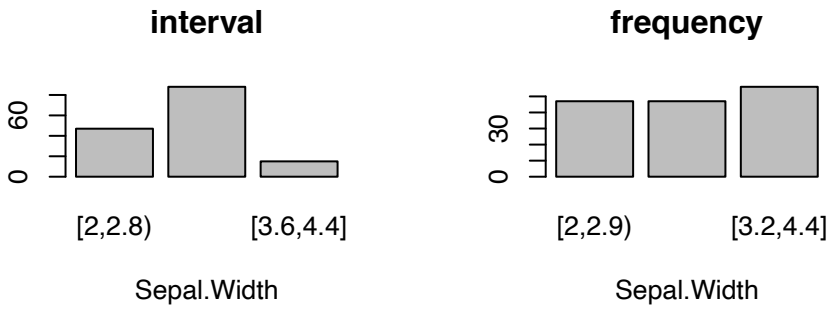
najbardziej przypomina normalny to Sepal.Width. Jest to jednoczesnie zmienna o najgorszych zdolnosciach dyskryminacyjnych. Natomiast zmienna, której rozklad prawie w ogóle nie przypomina normalnego to Petal.Length. Jest to zmienna o dobrych zdolnosciach dyksryminacyjnych. Sprawdzmy to rysujac wykres pudelkowy.

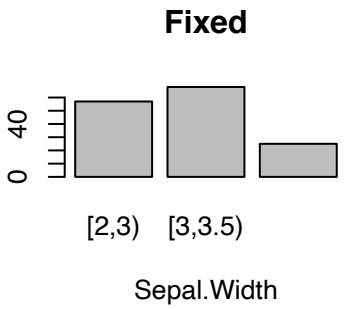
```
boxplot(Petal.Length~Species, col=1:3)
```



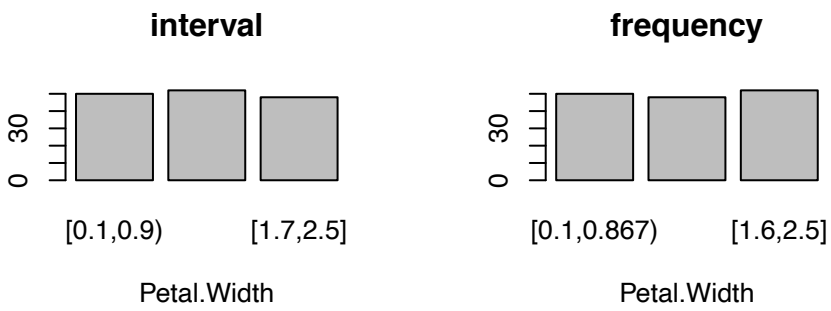
Omówmy teraz podstawowe metody dyskretyzacji

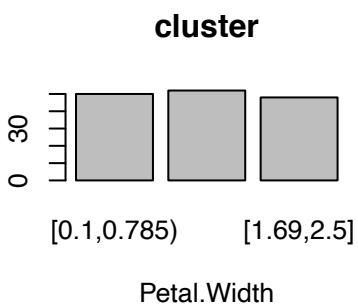
```
methods<-c(as.String("interval"),as.String("frequency"),as.String("cluster"))
par(mfrow=c(2,2))
for (j in c(1:3))
  plot(discretize(Sepal.Width, method=methods[j], breaks = 3),
       main=methods[j],
       xlab="Sepal.Width")
plot(discretize(Sepal.Width, method="fixed", breaks=c(min(Sepal.Width),3,3.5,
max(Sepal.Width))),main="Fixed",xlab="Sepal.Width")
```

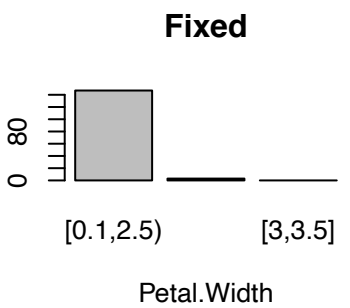





```
par(mfrow=c(2,2))
for (j in c(1:3))
  plot(discretize(Petal.Width, method=methods[j], breaks = 3),
       main=methods[j],
       xlab="Petal.Width")
plot(discretize(Petal.Width, method="fixed", breaks=c(min(Petal.Width),3,3.5,
max(Petal.Width))),main="Fixed",xlab="Petal.Width")
```

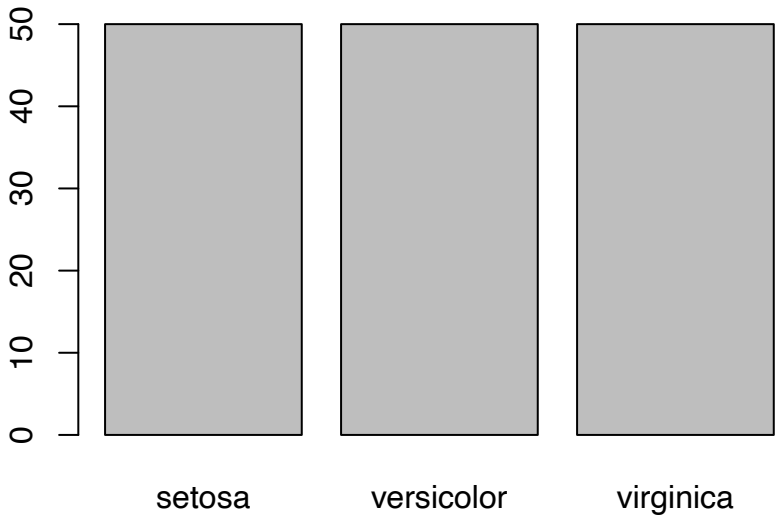






Metoda "Interval" dzieli przedzial zmiennej na równe czesci. Metoda "frequency" próbuje w kazdej kategorii umiescic tyle samo obserwacji. Metoda cluster dzieli obserwacje ze wzgledu na podobiestwo. Ostatnia metoda "fixed" pozwala uzytkownikowi wybrac zakres przedzialu w kategoriach. Wykresy dyskretyzacji dla zmiennej Sepal.Width sa bardzo rózne. Metody interval i cluster daja wykresy przyblizone do rozkladu normlanego Gdy spojrzymy na wykresy zmiennej Petal.Width rzuca sie w oko to, ze wykresy dla poszczegolnych metod prawie w ogóle sie nie róznia. Jedynie wykres dla metody Fixed odstaje od reszty, ale równie dobrze mozna by dobrac inne przedzialy. Ogólnie wyniki dla zmiennej o slabej zdolnosci dyskryminacyjnej sa calkowicie inne niz te, dla dobrej zmiennej.

```
plot(Species)
```



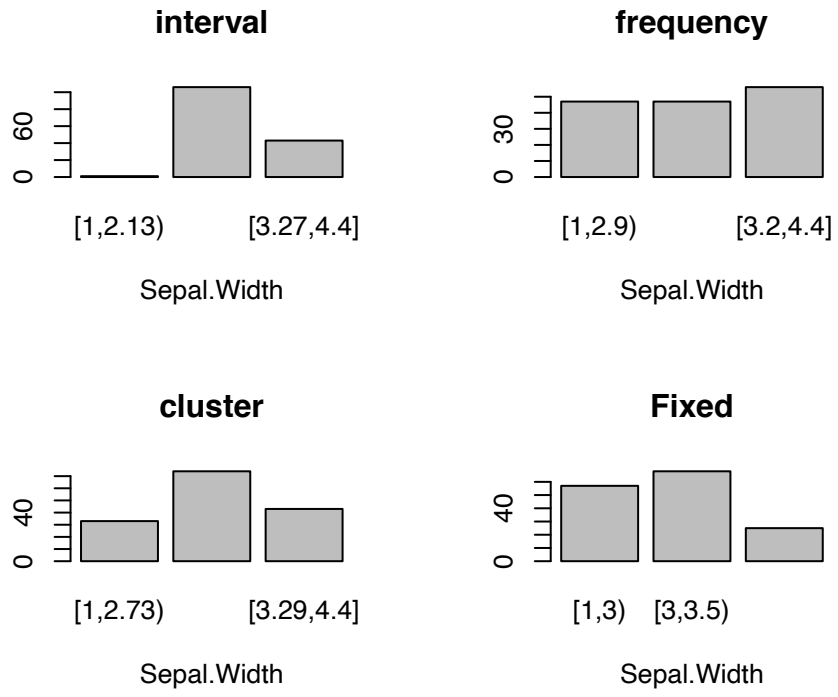
Gdy wezmiemy pod uwage rzeczywiste kategorie, najlepiej poradzila sobie metoda "frequency" dla zmiennej Sepal.Width Natomiast dla zmiennej Petal.Length wszystkie metody (oprócz fixed) poradzily sobie podobnie. Sprawdzmy nasze wnioski korzystajac z funkcji matchClasses() z pakietu e1071.

```
for (j in c(1:3))
  matchClasses(table(discretize(Petal.Width, method=methods[j], breaks = 3)
                     , Species))

## Cases in matched pairs: 96 %
## Cases in matched pairs: 94.67 %
## Cases in matched pairs: 66.67 %

for (j in c(1:3))
  matchClasses(table(discretize(Sepal.Width, method=methods[j], breaks = 3)
                     , Species))
```

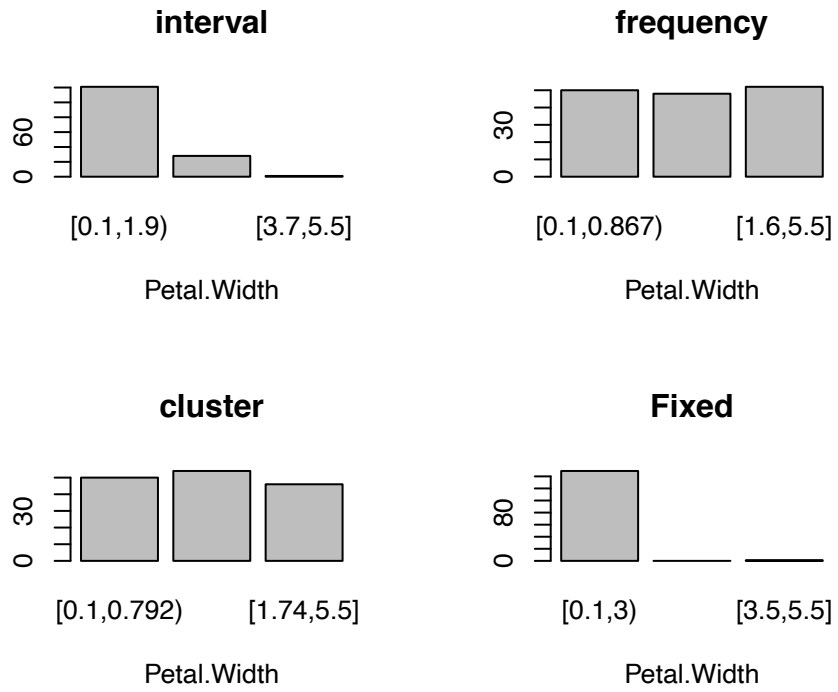
```
## Cases in matched pairs: 50.67 %
## Cases in matched pairs: 55.33 %
## Cases in matched pairs: 56 %
```

Jak widac wnioski z wykresów sa poprawne. Sprawdzmy czy dodanie wartosci odstajacych znacznie zmieni wyniki.

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

par(mfrow=c(2,2))
for (j in c(1:3))
  plot(discretize(Sepal.Width, method=methods[j], breaks = 3),
       main=methods[j],
       xlab="Sepal.Width")
plot(discretize(Sepal.Width, method="fixed",
                breaks=c(min(Sepal.Width),3,3.5,
max(Sepal.Width))),main="Fixed",xlab="Sepal.Width")
```

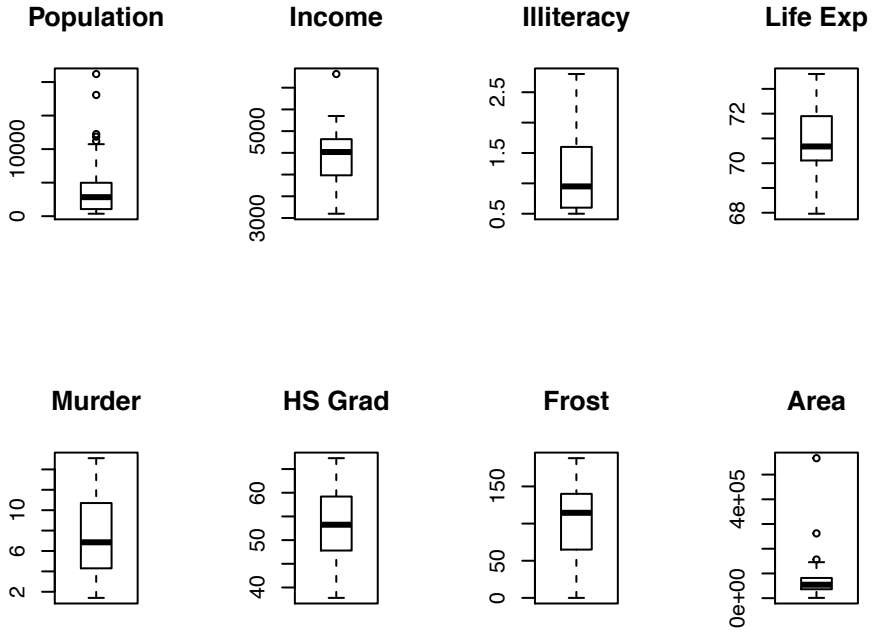



```
par(mfrow=c(2,2))
for (j in c(1:3))
  plot(discretize(Petal.Width, method=methods[j], breaks = 3),
       main=methods[j],
       xlab="Petal.Width")
plot(discretize(Petal.Width, method="fixed",
                breaks=c(min(Petal.Width),3,3.5,
max(Petal.Width))),main="Fixed",xlab="Petal.Width")
```



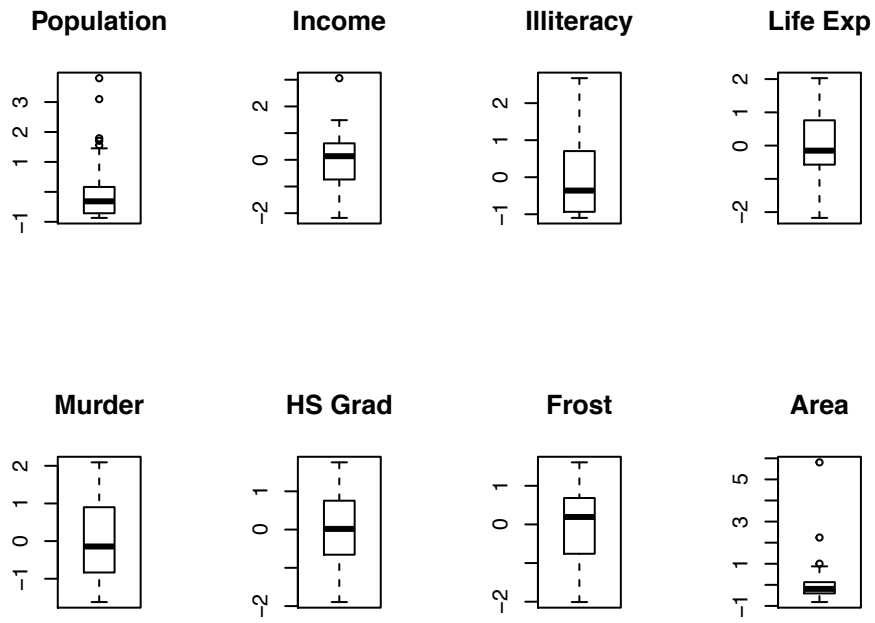
Jak mozna bylo sie spodziewac dodanie wartosci odstajacych znacznie pogorszy wyniki metody interval. Natomiast dla metody frequency nie ma to prawie zadnego znaczenia (o ile takich wartosci odstajacych dodamy malo). Dla metody cluster tez dodanie wartoci odstajacych nic nie zmienilo.

Zajmijmy sie teraz analiza danych state.x77. Wykorzystam do tego metode PCA redukcji wymiaru. Zacznijmy od wczytania danych i narysowania boxplotów.



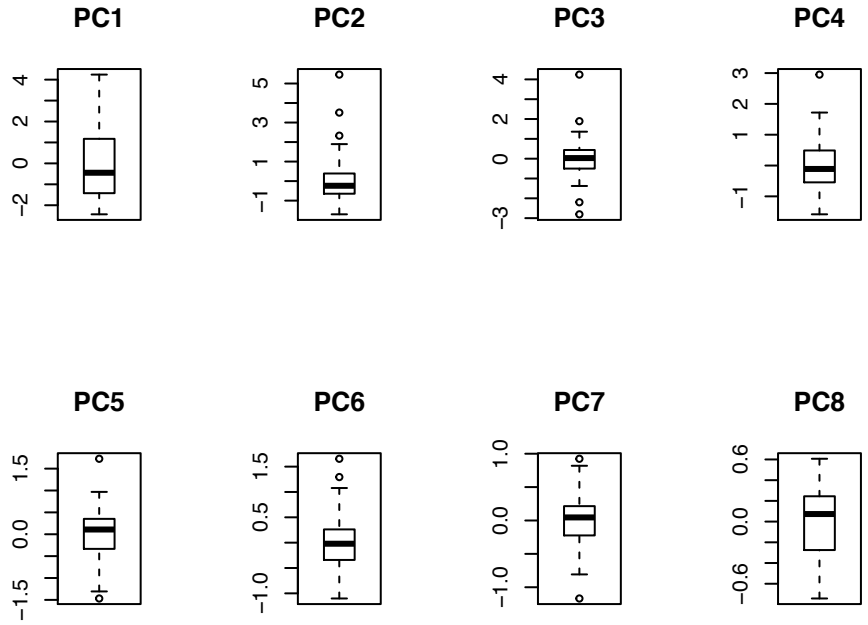
Jak widac zmienne znacznie sie róznia dlatego musimy je standaryzowac.

```
dane_scaled<-scale(dane)
par(mfrow=c(2,4))
for (i in c(1:8))
  boxplot(dane_scaled[,i],main=variables[i])
```



Zajmiemy sie teraz wyznaczaniem skladowych głównych. Wykorzystam do tego komende prcomp()

```
PC.data<-prcomp(dane_scaled)
scores = as.data.frame(PC.data$x)
attach(scores)
scores_indx<-names(scores)
par(mfrow=c(2,4))
for (i in c(1:8))
  boxplot(scores[i],main=scores_indx[i])
```




```
dane_scaled<-scale(dane)
```

Jak widac rozrzuty zmiennych znacznie róznia sie o siebie. Zaanlizujmy teraz wektory ladunków dla kilku pierwszych zmiennych.

```
PC.data2<-princomp(dane_scaled)
PC.data2$loadings
```

Loadings: Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Population 0.126 0.411 0.656 0.409 0.406 0.219 Income -0.299 0.519 0.100 -0.638 -0.462 Illiteracy 0.468 -0.353 -0.387 0.620 0.339 Life Exp -0.412 0.360 -0.443 0.327 -0.219 0.256 -0.527 Murder 0.444 0.307 -0.108 0.166 -0.128 0.325 0.295 -0.678 HS Grad -0.425 0.299 -0.232 0.645 0.393 0.307 Frost -0.357 -0.154 -0.387 0.619 0.217 -0.213 0.472 Area 0.588 -0.510 -0.201 0.499 -0.148 -0.286

Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 SS loadings 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 Proportion Var 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 Cumulative Var 0.125 0.250 0.375 0.500 0.625 0.750 0.875 1.000

```
summary(PC.data2)
```

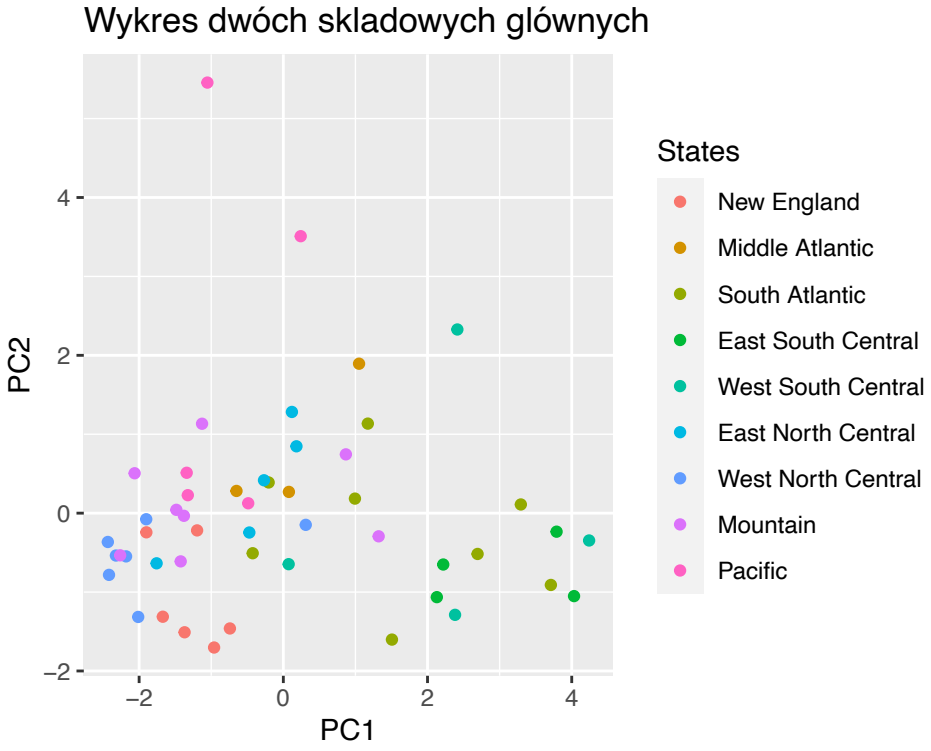
Importance of components: Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Standard deviation 1.8780090 1.2646268 1.0438881 0.83267888 0.61396161 Proportion of Variance 0.4498619 0.2039899 0.1389926 0.08843803 0.04808021 Cumulative Proportion 0.4498619 0.6538519 0.7928445 0.88128252 0.92936273 Comp.6 Comp.7 Comp.8 Standard deviation 0.54891933 0.3762443 0.33305246 Proportion of Variance 0.03843271 0.0180561 0.01414846 Cumulative Proportion 0.96779544 0.9858515 1.00000000

Dla trzech pierszych skladowych najwieksza wage maja zmienne odpowiednio Illiteracy, Area i Population. Warto zauwazyc, ze aby wyjasnic ok. 80 procent zmiennosci danych wystarcza

tylko trzy pierwsze skladowe. Jezeli jednak chcielibysmy otrzymac dokladniejsza analize, np. która wyjasnia 90 procent zmiennosci danych musimy juz wykorzystac az 5 skladowych.Ogólnie im dokladniejsze chcemy miec wyniki, tym wiecej skladowych musimy wykorzystac. Warto jednak zauwazyc, ze wyjasnienie dodatkowych 10 procent zmiennosci danych nie wplynie znacznie na rezultat, ale za to bardzo utrudni nam przebieg analizy (np. rysowanie wykresów).

Narysujmy teraz wykresy 2d i 3d pomiedzy skladowymi. state.name state.abb 1 Alabama AL 2 Alaska AK 3 Arizona AZ 4 Arkansas AR 5 California CA 6 Colorado CO 7 Connecticut CT 8 Delaware DE 9 Florida FL 10 Georgia GA 11 Hawaii HI 12 Idaho ID 13 Illinois IL 14 Indiana IN 15 Iowa IA 16 Kansas KS 17 Kentucky KY 18 Louisiana LA 19 Maine ME 20 Maryland MD 21 Massachusetts MA 22 Michigan MI 23 Minnesota MN 24 Mississippi MS 25 Missouri MO 26 Montana MT 27 Nebraska NE 28 Nevada NV 29 New Hampshire NH 30 New Jersey NJ 31 New Mexico NM 32 New York NY 33 North Carolina NC 34 North Dakota ND 35 Ohio OH 36 Oklahoma OK 37 Oregon OR 38 Pennsylvania PA 39 Rhode Island RI 40 South Carolina SC 41 South Dakota SD 42 Tennessee TN 43 Texas TX 44 Utah UT 45 Vermont VT 46 Virginia VA 47 Washington WA 48 West Virginia WV 49 Wisconsin WI 50 Wyoming WY

```
state.pca <- prcomp(state.x77, scale.=T, center=T, retx=T)
States<-kolory[as.numeric(state.division)]
p<-ggplot(scores,aes(x=PC1,y=PC2,color= state.division))
p<-p+geom_point()+ labs(color = "States",
                        title = "Wykres dwóch skladowych glównych")
p
```



Najbardziej wyrózniajace sie stany to East South Central i West South Central. Sprawdzmy co je charakteryzuje.

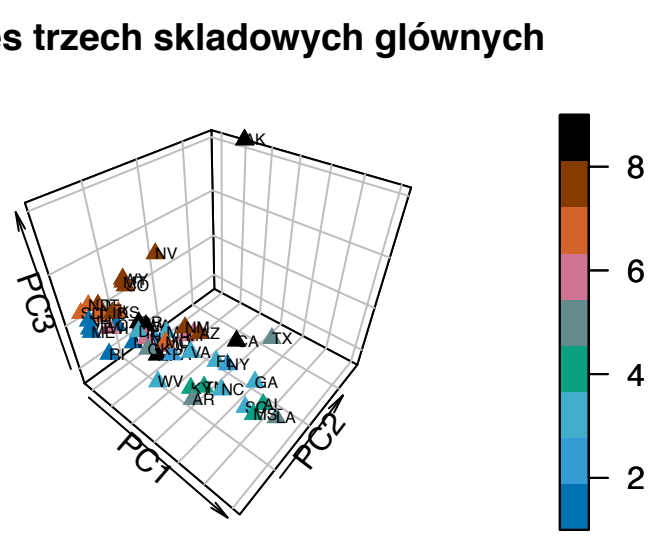
```
East_S_Ctr<-filter(data.frame(state.x77), state.division == 4)
West_S_Ctr<-filter(data.frame(state.x77), state.division == 5)
```

```
summary(East_S_Ctr)
summary(West_S_Ctr)
summary(state.x77)
```

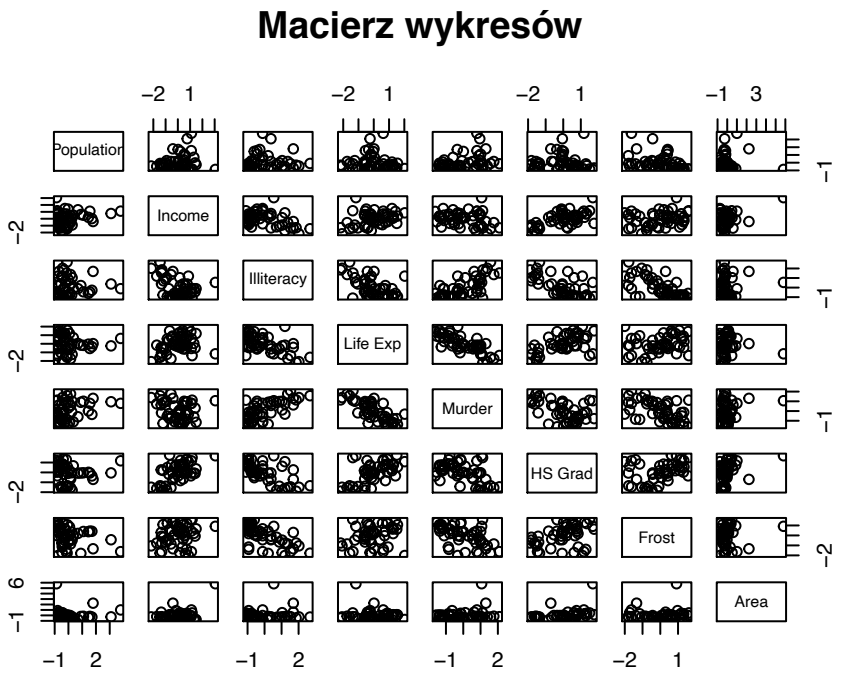
Porównujac podstawowe wskazniki sumaryczne tych dwóch stanów, ze wszystkimi stanami mozemy wyciagnac wniosek, ze East South Central i West South Central charakteryzuja:

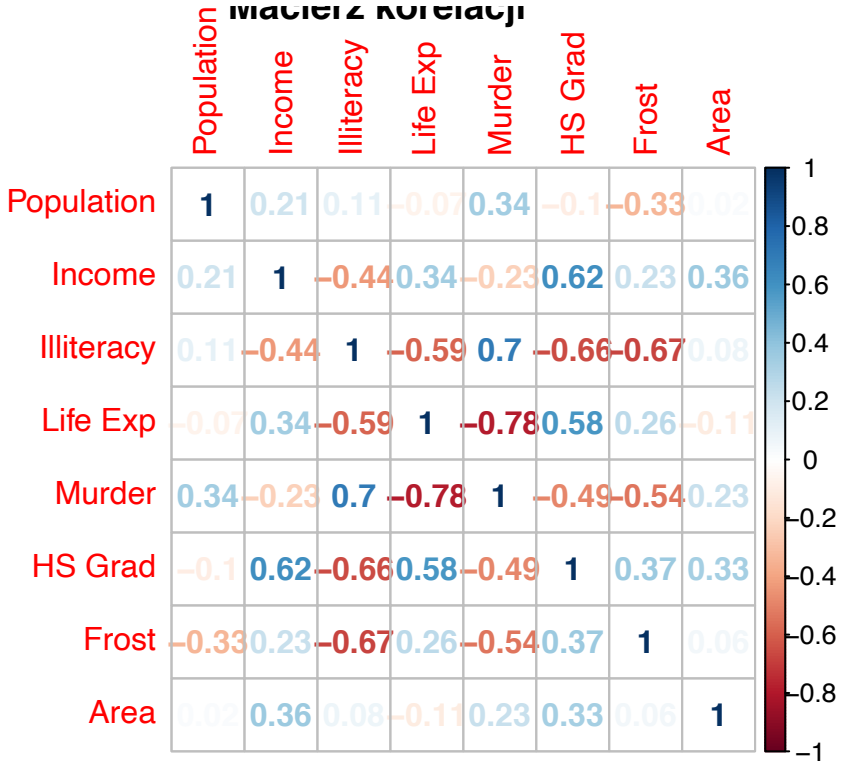
- niskie dochody,
- wysoki odsetek analfabetyzmu,
- wysoka liczba morderstw,
- niskie oceny w szkole sredniej.

```
scatter3D(PC1, PC2, PC3,  colvar=as.numeric(state.division), col=gg.col(9),
pch=17, xlab="PC1", ylab="PC2", zlab="PC3",
main="Wykres trzech skladowych glównych", bty = "b2")
text3D(PC1, PC2, PC3,  labels = state.abb, add=TRUE, cex=0.5)
```



Wykres 3d w sumie nie wnosi za wiele nowego. Teraz zajmijmy sie korelacja pomiedzy zmiennymi. Wykorzystam do tego funkcje corrplot() z pakietu corrplot i funkcje pairs, który tworzy macierz wykresów korelacji dla kazdej pary zmiennych.





Z oczywistych wniosków:

- wysokie oceny w szkole sredniej ida w parze z wysokimi zarobkami,
- niski poziom edukacji idze w parze z wysoka przestepczoscia,
- niski poziom edukacji jest ujemni skorelowany z zarobkami tak samo jak i z przewidywana dlugoscia zycia.
- niskie oceny w szkole sredniej.

Wazniejsze wnioski:

- analfabetyzm jest silnie skorelowany z morderstwami,
- srednia dlugosc zycia jest ujemnie skorelowana z morderstwami,

Ciekawy wniosek: Na podstawie przeprowadzonej analizy moznaby wyciagnac wniosek jakoby niska temperatura bylaby skorelowana z liczba przestepstw. Bez glebszej analizy oczywiscie ten wniosek jest absurdalny. To tez pokazuje pewna wadliwosc korelacji. Wiecej takich absurdalnych korelacji mozna znalezc na stronie http://tylervigen.com/spurious-correlations Ogólnie na podstawie tych danych mozna stworzyc prosty profil przestepcy. Wykresy 2 i 3d skladowych, informuja nas czy mozna pogrupowac wyniki ze wzgledu na podobiestwo obserwacji. Dla dwóch skladowych juz mozemy jak widac wyciagnac istotnie wnioski. Zastosowanie standaryzacji oczywiscie mialo pozytywny wplyw na wyniki analizy.

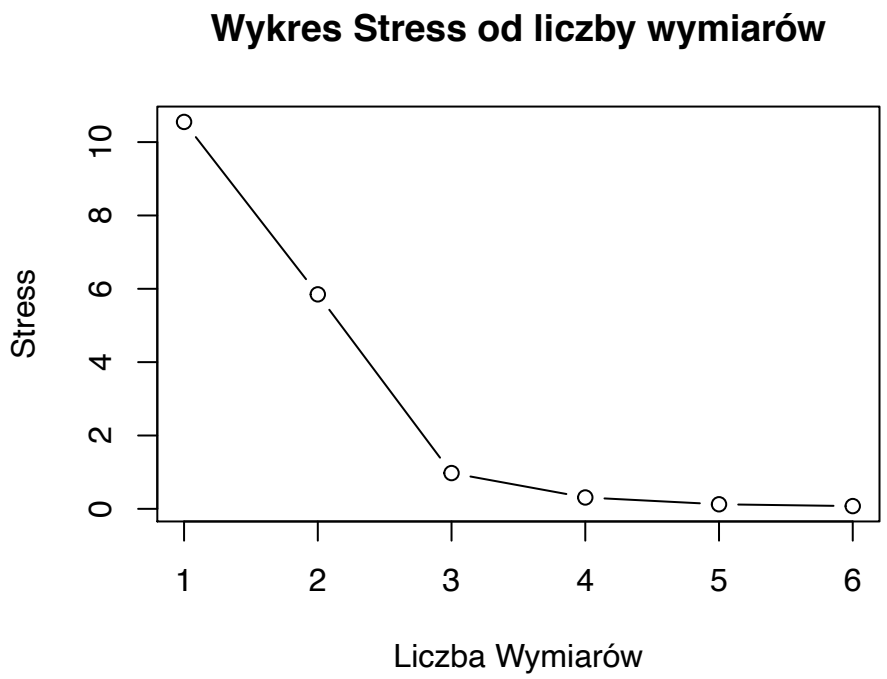
Teraz zajmijmy sie analiza danych iris z wykorzystaniem MDS.

```
data("iris")
data_iris <- subset(iris,select=-c(Species))
dane_scaled2<-scale(data_iris)
data_iris<-as.matrix(dane_scaled2)
dissimilarities<-dist(data_iris)
dis.matrix <- as.matrix(dissimilarities)
mds.results <- cmdscale(dis.matrix, k=3)
dis.matrix<-dis.matrix[-102,-103]
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="Liczba Wymiarów",main="Wykres Stress od liczby wymiarów")}
```

Tutaj korzystam z gotowego algorytmu do rysowania Stress w zaleznosci od wymiaru. W bibliografii podam odnosnik.

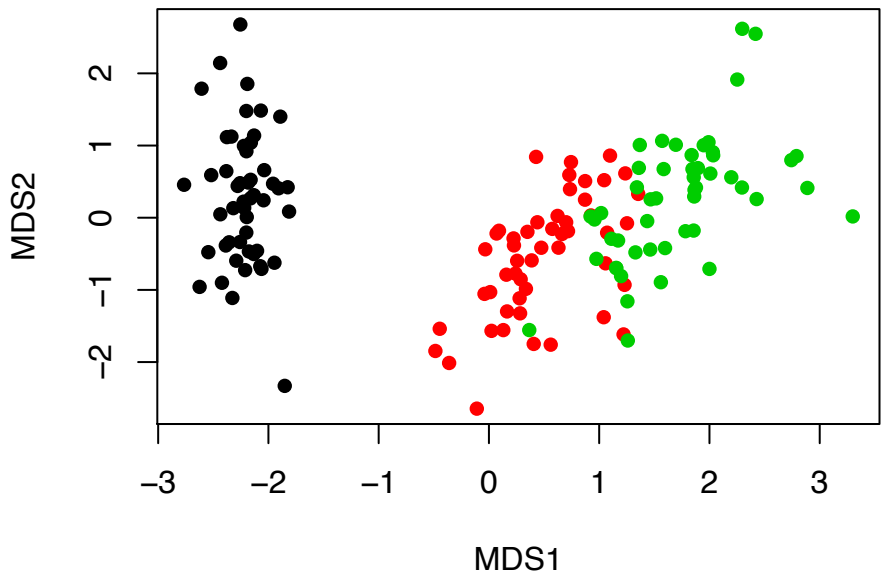
```
scree.plot(dist(dis.matrix),k=6)
```

initial value 0.213719 iter 5 value 0.128166 iter 10 value 0.108103 iter 15 value 0.096507 iter 20 value 0.089600 iter 25 value 0.085039 iter 30 value 0.081372 iter 35 value 0.078583 iter 40 value 0.076654 iter 45 value 0.075436 iter 50 value 0.074808 final value 0.074808 stopped after 50 iterations initial value 0.384448 iter 5 value 0.238243 iter 10 value 0.184211 iter 15 value 0.161924 iter 20 value 0.146573 iter 25 value 0.137533 iter 30 value 0.131674 iter 35 value 0.128477 iter 40 value 0.126292 iter 45 value 0.125068 iter 50 value 0.124274 final value 0.124274 stopped after 50 iterations initial value 1.164808 iter 5 value 1.008588 iter 10 value 0.948638 iter 15 value 0.872778 iter 20 value 0.709497 iter 25 value 0.580934 iter 30 value 0.469988 iter 35 value 0.390619 iter 40 value 0.348631 iter 45 value 0.330110 iter 50 value 0.309981 final value 0.309981 stopped after 50 iterations initial value 1.364044 iter 5 value 1.093667 iter 10 value 0.999851 iter 15 value 0.982641 final value 0.975711 converged initial value 7.247571 iter 5 value 6.430157 iter 10 value 6.099004 iter 15 value 5.870936 final value 5.848288 converged initial value 18.834169 iter 5 value 18.568632 iter 10 value 13.886079 iter 15 value 10.780324 final value 10.550075 converged

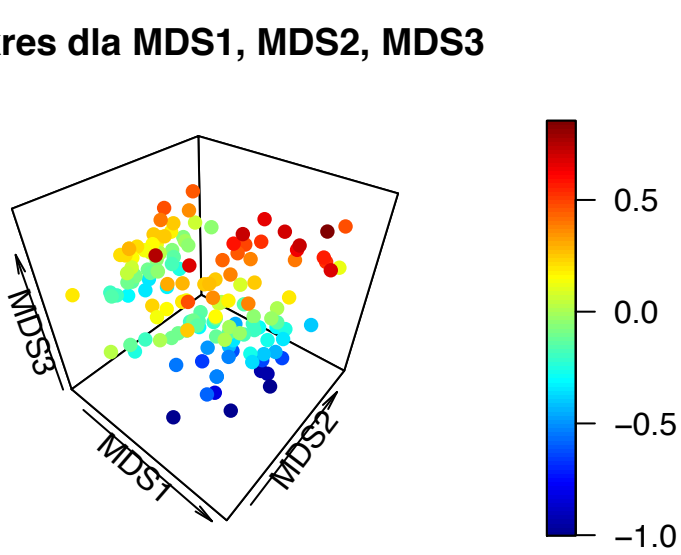


Optymalny jest wymiar 3, biorac pod uwage latwosc interpretacji wykresów. Wymiar 4 nie daje nam duzo lepszego rezultatu, a znacznie ciezej jest naryswac wykresy. Wizualizacja wykresów:

```
MDS1<-mds.results[,1]
MDS2<-mds.results[,2]
MDS3<-mds.results[,3]
plot(MDS1,MDS2, pch=16,col=Species)
```

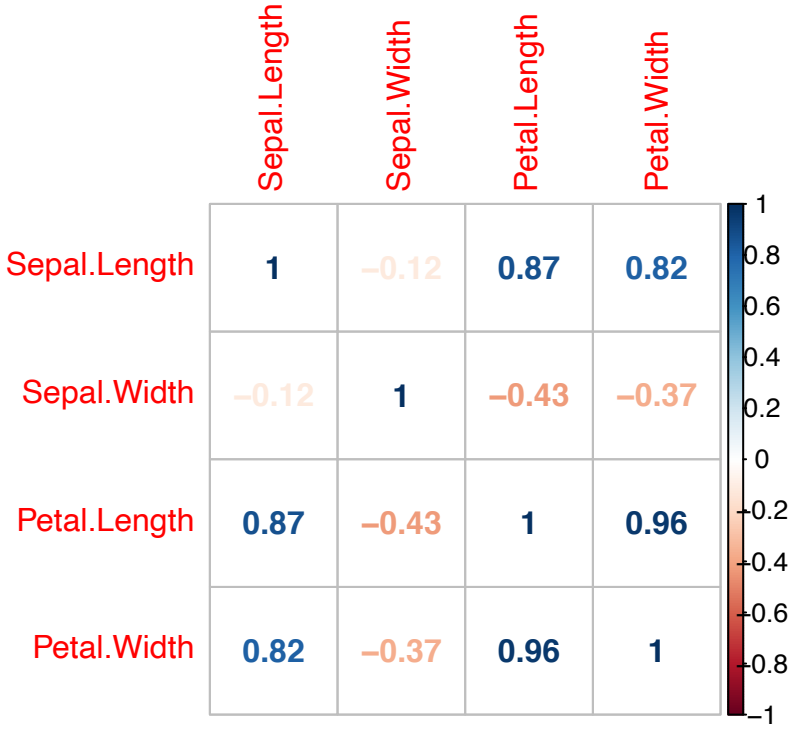


```
scatter3D(MDS1,MDS2,MDS3,pch=16,xlab="MDS1",ylab="MDS2",zlab="MDS3",main="Wykres dla MDS
```

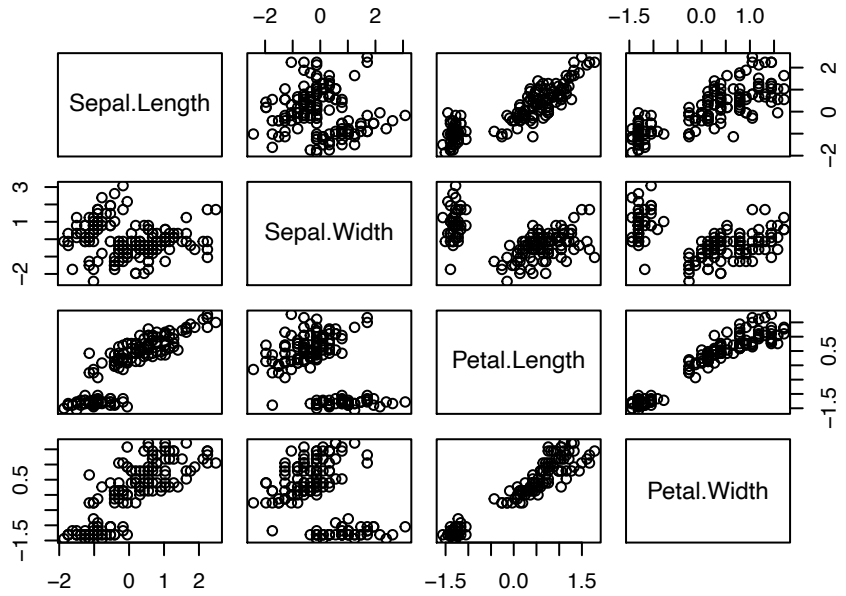


Tutaj w zasadzie nie ma ”interesujacych obiektów”. Wszystkie sie ladnie pogrupowaly, i zaden nie jest podejrzany. Z wykresów (szczególnie wykresu 2d) widac, ze uzyskalismy bardzo dobra separacje klas. Przeprowadzmy podobna analize jak dla poprzedniego zbioru danych.

Macierz korelacji

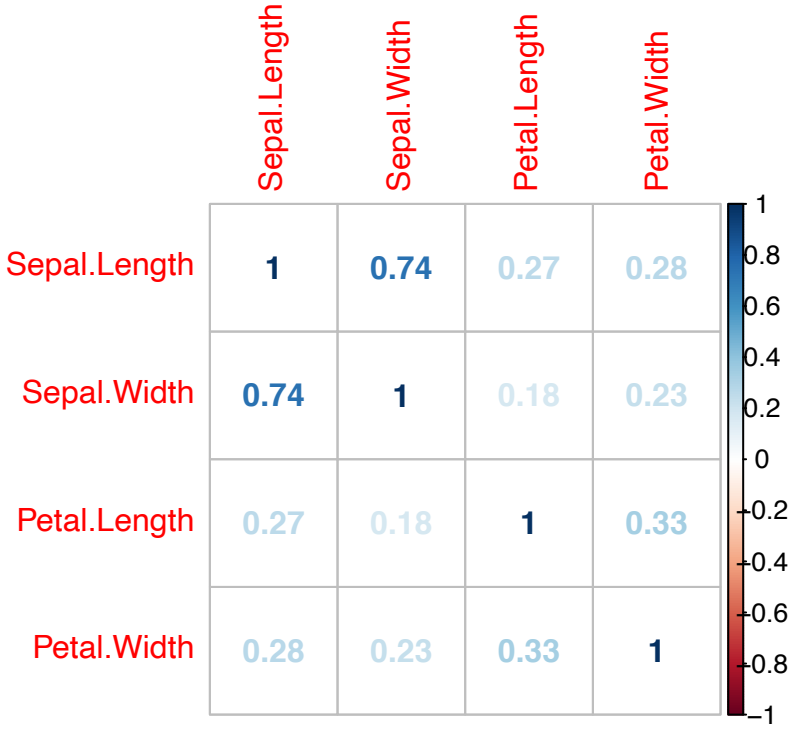


Macierz wykresów

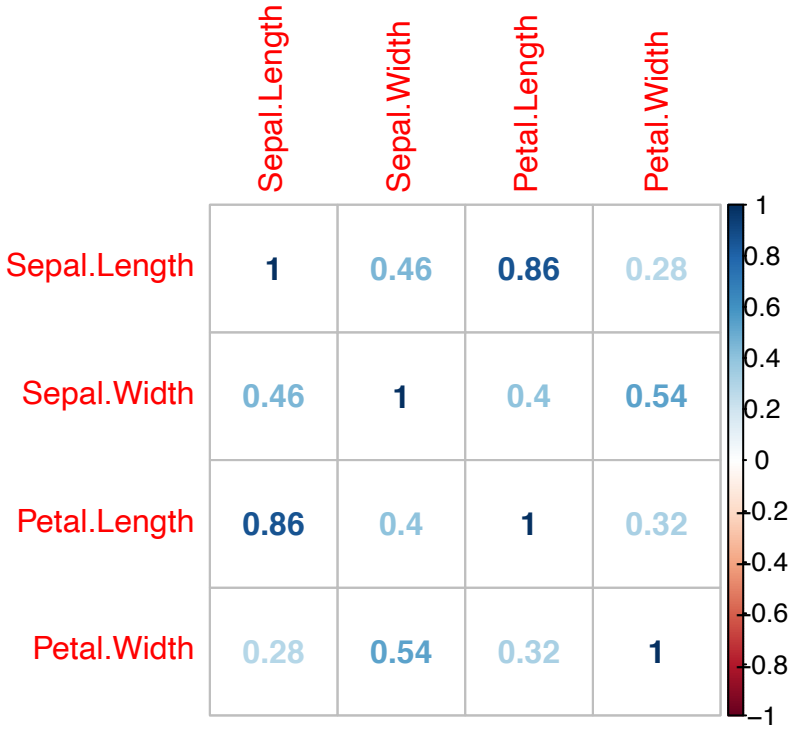


Wyniki korelacji dla klas które wyraznie sie od siebie róznia moga byc mylace. Przeprowadzmy zatem analize dla poszczególnych klas.

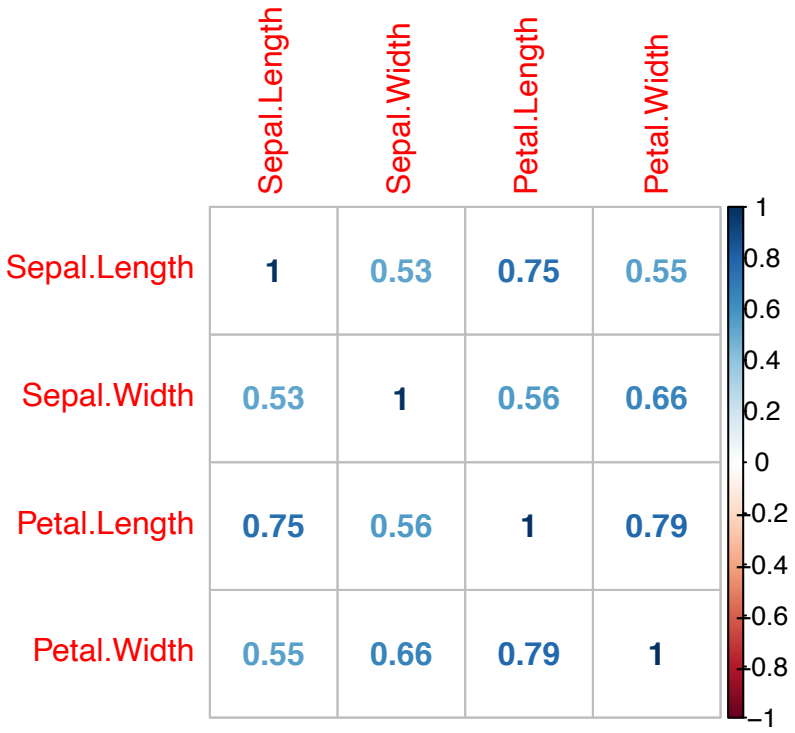
Macierz korelacji dla Setosa



Macierz korelacji dla Virginica



Macierz korelacji dla Versicolor



Ogólne wnioski:

- Dla gatunku Setosa jest pozytywna korelacja pomiedzy szerokoscia dzialki kielicha (Sepal), a jego dlugoscia. Pozostale parametry kwiatu sa ze soba slabo skorelowane. Mozemy wiec uznac, ze cecha wyrózniajaca gatunek Setosa to jednoczesnie dlugie i szerokie dzialki kielicha.
- Dla gatunku Virginica silnie skorelowane sa dlugosc dzialki kielicha i dlugosc platka. Pozostale parametry sa slabo skorelowane (chociaz korelacja na poziomoie 0.54 dla zmiennych Sepal i Petal width jest juz znaczaca). Mozemy wiec wysunac wniosek, ze gatunek Virginica wyróznia przede wszystkim dlugie platki i dlugie dzialki kielicha. Ale znajdziemy równiez okazy, które maja szerokie platki i szerokie dzialki kielicha.
- Dla gatunku Versicolor widzimy, ze wiele zmiennych jest ze soba skorelowanych. Jest to gatunek, który na pewno bedzie ciezej bedzie odróznic od gatunku Virginica niz od gatunku Seposa. Jako jedyny z tych gatunków ma jednoczesnie szerokie jak i dlugie platki.

4 Podsumowanie

Podsumowanie metody MDS i PCA okazuja sie bardzo przydatne w analizie danych. Dzieki nich moze precyzjnie wysunac wnioski i wskazac zaleznosci pomiedzy zmiennymi.

Literatura

[1] Stackoverflow, https://stackoverflow.com

[2] Data mining course site, http://prac.im.pwr.wroc.pl/~zagdan/polish_ver/ED2020/index.html

[3] Algorytm rysowania Stress od liczby wymiarów, https://rpubs.com/YaPi/393252