Podejścia Systemowe w Badaniach Biomedycznych
zajęcia laboratoryjne 2

Redukcja wymiarowości

1 Redukcja wymiarowości

Redukcja wymiarowości jako ważny element eksploracji danych jest to proces transformacji danych wielowymiarowych do przestrzeni o mniejszym wymiarze. Podejście to jest stosowane ze względów praktycznych, celem ułatwienia wizualizacji danych. Najczęściej dane są redukowane są do 2 lub 3 wymiarów. Dodatkowo, po zmniejszeniu wymiaru, wiele zależności wydaje się być bardziej czytelnych.

Celem metod redukcji danych jest zidentyfikowanie ukrytego wzorca danych, zmniejszenie wymiarowości poprzez usunięcie szumów i redundantnych danych oraz zidentyfikowanie skorelowanych zmiennych.

W R dostępnych jest wiele różnych funkcji i pakietów dla przeprowadzenia redukcji wymiarowości różnymi metodami. Wyniki mogą się różnić pomiędzy tymi pakietami. W tym protokole skupimy się tylko na wybranych z nich.

2 Ocena danych wejściowych

Często otrzymane dane wejściowe, bez wcześniejszego przetwarzania nie nadają się do bezpośredniego użycia i przeprowadzenia analiz. W danych mogą pojawiać się braki i wartości odstające, które mogą wpływać na wyniki analiz. Opis tych właściwości i sposoby radzenia sobie z nimi zostały omówione na poprzednich zajęciach i wykładzie.

Należy jednak wspomnieć także o normalizacji i skalowaniu danych. Konieczność takiej obróbki wynika z różnego charakteru danych. Jeśli zmienne charakteryzują się różną skalą to ważne jest zapewnienie porównywalności tych zmiennych. R w niektórych funkcjach domyślnie wykonuje takie przetwarzanie danych. Wyróżniamy następujące przekształcenia danych:

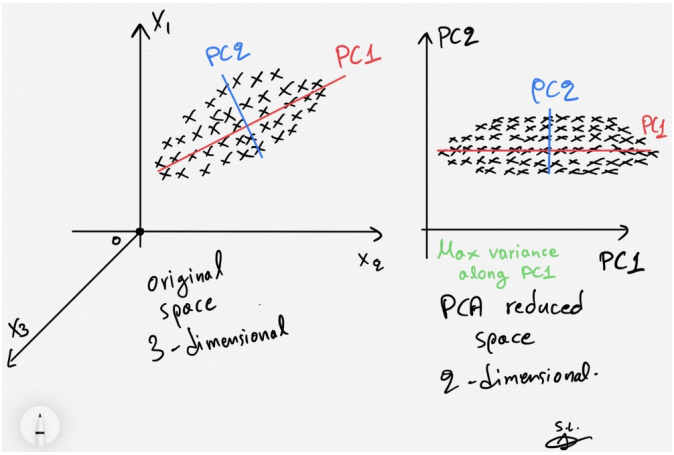
- normalizacja min-max: dane po normalizacji należą do przedziału od 0 do 1,
- standaryzacja: dane po standaryzacji będą miały średnią = 0 i odchylenie standardowe = 1,
- centrowanie: centrowanie danych wokół wartości 0.

3 Analiza składowych głównych - PCA

Analiza składowych głównych (*ang. Principal Components Analysis - PCA*) pozwala podsumować i wizualizować informacje w zbiorze danych, które zawierają obserwacje opisane przez wiele wzajemnie skorelowanych zmiennych ilościowych (każda zmienna może zostać uznana za inny wymiar). Zastanów się, jak wizualizować dane jeśli mamy więcej niż trzy zmienne? Analiza składowych głównych służy do wyodrębnienia ważnych informacji z wielowymiarowych danych i wyrażenia tych informacji jako zestawu kilku nowych zmiennych zwanych właśnie składowymi głównymi (*ang. Principal Components - PC*), takie nowe zmienne odpowiadają liniowej kombinacji oryginałów. Celem analizy PCA jest znalezienie transformacji układu współrzędnych, która lepiej opisze zmienność pomiędzy obserwacjami. Innymi słowy, chodzi o zidentyfikowanie kierunków (składowych głównych), wzdłuż których zmienność danych jest maksymalna. Przy redukcji wymiarowości danych zależy nam na ograniczeniu ilości wielowymiarowych danych do dwóch lub trzech składowych głównych, przy minimalnej utracie informacji.

3.1 Zrozumieć PCA

Zrozumienie szczegółów PCA wymaga znajomości algebry liniowej. Najprościej jednak można wyjaśnić podstawy na bazie prostej graficznej reprezentacji danych. Na poniższym wykresie 1 po lewej stronie, dane są przedstawione w przestrzeni 3D. Zmniejszenie wymiarów uzyskuje się poprzez określenie głównych kierunków, zwanych składowymi głównymi, w których dane się zmieniają. Na rysunku 1 po lewej stronie, oś PC1 jest pierwszym głównym kierunkiem, wzdłuż którego próbki wykazują największą zmienność. Oś PC2 jest drugim najważniejszym kierunkiem i jest prostopadła do osi PC1. Wymiarowość 3D można zredukować do wymiaru 2D, co przedstawia wykres 1 po prawej stronie.



Rysunek 1: Zrozumienie koncepcji PCA na przykładzie redukcji wymiaru 3D do 2D.
Źródło: https://ai.plainenglish.io/how-to-implement-pca-with-python-and-scikit-learn-22f3de4e5983

Jak to rozumieć od strony technicznej? Za pomocą pewnych transformacji liniowych, tworzony jest nowy układ współrzędnych, gdzie pierwsza oś ma największą wariancję, a druga oś ma drugą w kolejności największą wariancję. Składowe, których wariancja jest niska uznane są za nieważne i można je pominąć. Nowo utworzony zbiór zmiennych, ma być odpowiednio mały, ale tak dobrany aby odzwierciedlić zmienność występującą w oryginalnym zestawie danych. Wielkość wariancji zachowana przez każdą składową główną jest mierzona tak zwaną wartością własną (*ang. eigenvalue*).

3.2 Przykład *prcomp()* i *princomp()* w R

Najbardziej podstawowymi funkcjami do analizy PCA jest *prcomp()* i *princomp()* z pakietu *stats*, jednak każda z tych funkcji wykorzystuje inną metodę:

- *prcomp()* - wyznacza składowe główne przy użyciu dekompozycji SVD,
- *princomp()* - wyznacza składowe główne poprzez wektory własne macierzy kowariancji/korelacji.

W obu powyższych przypadkach należy podać dane wejściowe i określić pozostałe argumenty. W przypadku funkcji *prcomp* należy określić czy dane mają zostać przeskalowane przed przystąpieniem do analizy, służy do tego argument *scale* (*scale = TRUE* jeśli dane mają być skalowanie lub *scale = FALSE* jeśli nie). Podobnie jest w przypadku funkcji *princomp()*, ustawienie argumentu *cor = TRUE* określa, że dane mają zostać przeskalowane i centrowane przed analizą. Opisane wyżej metody teoretycznie dają identyczne wyniki (właściwości składowych głównych będą identyczne), jednak w niektórych sytuacjach wyniki mogą się różnić.

Dane wykorzystywane do poniższych dane pochodzą z pakietu *factoextra*:

```
1 help(package="factoextra") #zobacz co oferuje pakiet
2 install.packages("factoextra")
3 library("factoextra")
4
```

```r
data(decathlon2) #dane z pakietu factoextra
dane <- decathlon2[1:10]
```

Przykład użycia funkcji *prcomp()*:

```r
#analiza PCA
result.prcomp <- prcomp(dane, scale = TRUE, center = TRUE)
result.prcomp

#wykres obserwacji (individuals/x/scores)
result.prcomp$x
plot(result.prcomp$x, main = "scale = TRUE & center = TRUE")
text(result.prcomp$x, labels = row.names(dane), pos = 3, cex = 0.7)

#wykres zmiennych (variables/rotation/loadings)
rotation <- result.prcomp$rotation
rotation
plot(rotation, main = "scale = TRUE & center = TRUE")
text(rotation, labels = colnames(dane), pos = 2, cex = 0.7)
for(i in 1:dim(dane)[2]) {
  arrows(0, 0, rotation[i, 1], rotation[i, 2], angle = 7, col = "red")
}

#wykres obserwacji i zmiennych
biplot(result.prcomp)

#wykres osuwiskowy - wykres wariancji
screeplot(result.prcomp, main = "scale = TRUE & center = TRUE", type = "l")

#wykres osuwiskowy mozna takze otrzymac przy uzyciu funkcji plot()
plot(result.prcomp, main = "scale = TRUE & center = TRUE")

#podsumowanie danych przedstawionych na wykresie screeplot() i plot()
summary(result.prcomp)
```

Przykład użycia funkcji *princomp()*:

```r
#analiza PCA: cor = T macierz korelacji, cor = F macierz kowariancji
result.princomp <- princomp(dane, cor = TRUE)
result.princomp

#wykres obserwacji (individuals/x/scores)
result.princomp$scores
plot(result.princomp$scores)
text(result.princomp$scores, labels = row.names(dane), pos = 3, cex = 0.7)

#wykres zmiennych (variables/rotation/loadings)
laoding <- result.princomp$loadings
laoding
plot(laoding)
text(laoding, labels = colnames(dane), pos = 2, cex = 0.7)
for(i in 1:dim(dane)[2]) {
  arrows(0, 0, laoding[i, 1], laoding[i, 2], angle = 7, col = "red")
}

```

```r
#wykres obserwacji i zmiennych
biplot(result.princomp)

#wykres osuwiskowy
screeplot(result.princomp, type = "l")

#wykres osuwiskowy mozna takze otrzymac przy uzyciu funkcji plot()
plot(result.princomp)

#podsumowanie danych przedstawionych na wykresie screeplot() i plot()
summary(result.princomp)
```

W kontekście wyników uzyskanych z funkcji *prcomp* i *princomp* należy zwrócić uwagę na inne nazwy obiektów, które w rzeczywistości wskazują te same wyniki:

- zmienna *x* dla *prcomp()* i zmienna *scores* dla *princomp()*: współrzędne obserwacji składowych głównych,
- zmienna *rotation* dla *prcomp()* i zmienna *loadings* dla *princomp*(): wektory ładunków.

3.3 Pakiet *factoextra*

Pakiet *factoextra* został stworzony by umożliwiać wydobywanie danych i lepszą wizualizację dla różnych metod redukcji wymiaru, poniżej zapoznamy się z wybranymi funkcjami.

Wydobywanie danych:

```r
#wyniki uzyskane z wczesniejszej analizy
result.prcomp

#otrzymanie wartosci wlasnych i wariancji skladowych glownych
get_eigenvalue(result.prcomp)

#otrzymanie wynikow dla obserwacji
get_pca_ind(result.prcomp)
#Principal Component Analysis Results for individuals
#===================================================
#  Name       Description
#1 "$coord"   "Coordinates for the individuals"
#2 "$cos2"    "Cos2 for the individuals"
#3 "$contrib" "contributions of the individuals"
get_pca_var(result.prcomp)$coord
get_pca_var(result.prcomp)$cos2 #jakosc reprezentacji
get_pca_var(result.prcomp)$contrib #udzial w skladowych glownych

#otrzymanie wynikow dla zmiennych
get_pca_var(result.prcomp)
```

Jaka jest interpretacja powyższych wyników?

- **Wartości własne i wariancje składowych głównych**: w wyniku działania funkcji *get_eigenvalue()*, otrzymujemy tabelę z trzema kolumnami (*eigenvalue*, *variance.percent*, *cumulative.variance.percent*). Pierwsza kolumna zawiera wartości własne, suma wszystkich tych wartości daje całkowitą wariancję równą 10. Druga kolumna zawiera procent wariancji, który oznacza że $X_1\%$ zmienności jest wyjaśnione przez konkretną wartość własną. Trzecia kolumna zawiera skumulowany procent wariancji (uzyskany przez sumę kolejnych procentów wariancji ($X_1\% + X_2\% + ...$). Załóżmy, że mowa o sumie dwóch pierwszych procentów wariancji (dla 2 pierwszych wartości własnych), zatem suma $X_1\% + X_2\%$ jest to procent zmienności wyjaśniony przez dwie pierwsze wartości własne (wartości własne określają poszczególne składowe główne).

Sprawdzamy wartości własne, aby określić liczbę składowych głównych, które mają zostać zachowane po analizie PCA. Wybierane są te, które mają największe wartości, co ma na celu minimalizację straty informacji podczas redukcji wymiarów. Wartości własne są wyższe dla pierwszych składowych głównych i niższe dla kolejnych. Wartości własne są miarą zmienności pierwotnych danych przedstawionych we współrzędnych składowych głównych.

- **Wyniki dla obserwacji i zmiennych**: funkcja *get_pca_ind()* wydobywa dane dla obserwacji, podczas gdy funkcja *get_pca_var()* wydobywa informacje dla zmiennych. Poniżej znajduje się lista obiektów, przez które możemy dostać się do szczegółowych danych:
  - *coord* - wydobywa współrzędne zmiennych wykorzystywane w celu utworzenia wykresu punktowego,
  - *cor* - wydobywa dane o korelacjach między zmiennymi a wymiarami,
  - *cos2* - wydobywa dane o jakości reprezentacji zmiennych,
  - *contrib* - wydobywa dane o udziale procentowym zmiennych w składowych głównych.

  Uzyskane dane wykorzystuje się na dalszym etapie do wizualizacji.

Wizualizacja danych:

```
#Wizualizacja wartosci wlasnych na wykresie osuwiskowym
fviz_eig(result.prcomp)
fviz_eig(result.prcomp, addlabels = TRUE)

#wykres obserwacji (individuals/x/scores)
fviz_pca_ind(result.prcomp)
#dodatkowe mozliwosci reprezentacji obserwacji
fviz_pca_ind(result.prcomp,
  col.ind = "cos2", #kolor wedlug jakosci reprezentacji
  repel = TRUE #nie nakladaja sie nazwy
)

#wykres jakosci reprezentacji
fviz_cos2(result.prcomp, choice = "ind", axes = 1:2) #2 wymiary
#fviz_cos2(result.prcomp, choice = "ind") # 1 wymiar

#wykres udzialu w skladowych glownych
fviz_contrib(result.prcomp, choice = "ind", axes = 1:2) #2 wymiary

#wykres zmiennych (variables/ratation/loadings)
fviz_pca_var(result.prcomp)
#dodatkowe mozliwosci reprezentacji obserwacji
fviz_pca_var(result.prcomp,
  col.var = "contrib", #kolor wedlug udzialu w skladowych glownych
  repel = TRUE
)

#wykres jakosci reprezentacji
fviz_cos2(result.prcomp, choice = "var", axes = 1:2) #2 wymiary

#wykres udzialu w skladowych glownych
fviz_contrib(result.prcomp, choice = "var", axes = 1:2) #2 wymiary

#biplot
fviz_pca_biplot(result.prcomp, repel = TRUE)
```

Jaka jest interpretacja powyższych wyników?

- **Wykres osuwiskowy**: wykres osuwiskowy utworzony za pomocą funkcji *fviz_eig()* przedstawia procent wariancji dla każdego wymiaru. Można użyć dodatkowego argumentu *addlabels = TRUE*, aby dodać etykiety z dokładnymi wartościami. Wykres ten jest graficzną wizualizacją zmienności, można określić ile procent zmienności pierwotnych danych wyjaśnia poszczególna składowa główna.

- **Wykres dla zmiennych**: podstawowy wykres dla zmiennych można utworzyć funkcją *fviz_pca_var()*. Wykres dla zmiennych pokazuje wpływ zmiennych na poszczególne składowe główne. Zmienne są reprezentowane przez wektory (zwane wektorami ładunków), a ich kierunek i długość określają w jakim stopniu wpływają na poszczególne składowe. Gdy zmienne leżą blisko siebie są silnie dodatnio skorelowane, gdy leżą po przeciwnych stronach są silnie ujemnie skorelowane, a gdy leżą względem siebie prostopadle to nie wykazują żadnej korelacji. Ponadto, w przypadku zmiennych na wykresie może pojawić się tzw. koło korelacji: im większa wartość to zmienne ułożone są bliżej obwodu koła korelacji, a im niższa tym zmienne ułożone są bliżej środka tego koła i są mniej ważne dla składowych głównych - odległość zmiennych jest miarą jakości.

  Dla wykresu zmiennych można użyć dodatkowych argumentów jak *col.var*, żeby otrzymać dodatkową informację kolorystyczną, np. o jakości reprezentacji (col.var = "cos2") czy o udziale w składowych głównych (col.var = "contrib"). Wspomniane dane, można narysować także na oddzielnych wykresach:

  - wykres jakości reprezentacji *fviz_cos2()* z argumentem choice = "var": wysoka wartość świadczy o dobrej reprezentacji zmiennej w składowej głównej.
  - wykres udziału w składowych głównych *fviz_contrib()* z argumentem choice = "var": im większa wartość, tym większy udział zmiennej w składowej głównej. Czerwona przerywana linia na tym wykresie wskazuje oczekiwany średni wkład, zmienne przekraczające ten próg mają największy udział w badanych składowych głównych.

- **Wykres dla obserwacji**: podstawowy wykres dla obserwacji można utworzyć funkcją *fviz_pca_ind()*. Wykres ten pokazuje, które grupy obserwacji są do siebie podobne (są zgrupowane razem), a które się od siebie różnią (są od siebie oddalone). Podobnie jak dla wykresu zmiennych, również dla wykresu obserwacji można użyć dodatkowych argumentów jak *col.ind*, żeby otrzymać dodatkową informację kolorystyczną, np o jakości reprezentacji (col.ind = "cos2") czy o udziale w składowych głównych (col.ind = "contrib"). Wspomniane dane, można narysować także na oddzielnych wykresach (podobnie jak dla zmiennych, należy tylko ustawić odpowiednio argumenty):

  - wykres jakości reprezentacji *fviz_cos2()* z argumentem choice = "ind",
  - wykres udziału w składowych głównych *fviz_contrib()* z argumentem choice = "ind".

- **Wykres obserwacji i zmiennych**: funkcja *biplot()* należy zdawać sobie sprawę, że obserwacje i zmienne nie leżą w tej samej przestrzeni, dlatego powinno się skupiać na kierunku, a nie na pozycjach bezwzględnych na wykresie. Im dana obserwacja leży bliżej od danej zmiennej, tym większy jest wpływ tej zmiennej na daną obserwację.

Zadania

Sprawozdanie i kod R prześlij na maila (Kaja.Gutowska@cs.put.poznan.pl). Proszę nadać tytuł maila zgodnie z opisem: PSwBB_Imię_Nazwisko_lab2.

Zad. 1

Przetestuj funkcję *prcomp()* i *princomp()*. Zaobserwuj jakie są wyniki przy zastosowaniu skalowania i centrowania oraz jak wyglądają dane gdy nie zastosujemy transformacji danych. Pamiętaj, że obie te metody używają innych argumentów. Jakie wartości, przy jakich argumentach pozwalają uzyskać takie same wyniki i wykresy? Jakie są konsekwencje, gdy nie przeprowadzimy skalowania danych?

Zad. 2

Wybierz i wykorzystaj istniejące dane np. ze strony Głównego Urzędu Statystycznego (https://stat.gov.pl) lub dowolne inne dane, aby przeprowadzić redukcję wymiarowości metodą PCA. Opisz wybrane dane, poszczególne etapy analizy i uzyskane wyniki.