AwAD - Projekt 3

Piotr Wysocki, Mikołaj Bójski,
Sebastian Botero Leonik, Aleksander Luckner

1 Motywacja

Pracując z zestawami danych, często chcemy je zwizualizować na wykresie, albo po prostu je opisać, w czym może nam przeszkodzić za duża liczba kolumn tabeli. Przyjmijmy, że nasze dane są w tabeli, która ma N wierszy i M kolumn. Wtedy można potraktować tą tabelę jako N-elementowy zbiór wektorów z przestrzeni $\mathbb{R}^M$. Tu od razu rzuca nam się problem z zaznaczeniem tych punktów na wykresie: Jest to możliwe tylko dla $M \leqslant 3$, a przejrzyste tylko dla $M \leqslant 2$. Tym samym rodzi się potrzeba zmniejszenia wymiarowości naszych danych.

2 Wyprowadzenie

Ze względów wygody notacji, tu komórkę macierzy A w n-tym wierszu i m-tej kolumnie oznaczamy a_n^m

Oznaczmy D - macierz będąca tabelą danych, która jako wiersze ma wektory d_i^T

$$D = \begin{bmatrix} d_1^T \\ \vdots \\ d_N^T \end{bmatrix}$$

Najprostszym sposobem na zmniejszenie wymiaru naszych danych jest zrzutowanie ich na przystrzeń niżej wymiarową.

Na początku spróbujmy rozwiązać problem zrzutowania na przestrzeń jednowymiarową. Aby jak najlepiej zachować informację zawartą w danych i nie komplikować zbędnie obliczeń, przyjmijmy, że będzie to rzut ortogonalny. Tym samym nasze zadanie sprowadza się do znalezienia odpowiedniej prostej l, na którą zrzutujemy dane.

Aby łatwiej taką prostą wyznaczyć, wypośrodkujmy nasz zestaw danych, poprzez odjęcie od każdego wektora środek ciężkości. To przesunięcie sprawi, że sensowne będzie szukanie l wśród prostych przechodzących przez środek układu współrzędnych. Oznaczmy

$$\bar{d} = \frac{1}{N}\sum_{i=1}^{N} d_i \qquad X = D - \begin{bmatrix} \bar{d}^T \\ \vdots \\ \bar{d}^T \end{bmatrix} = \begin{bmatrix} (d_1 - \bar{d})^T \\ \vdots \\ (d_N - \bar{d})^T \end{bmatrix} = \begin{bmatrix} (x_1)^T \\ \vdots \\ (x_N)^T \end{bmatrix}$$

Potraktujmy l teraz jako przestrzeń liniową generowaną przez wektor jednostkowy v, $v^T v = 1$. Zauważmy, że jeśli dana x_i jest prostopadła do v to należy ona do jądra rzutu na $\mathcal{L}(v)$ i wtedy tracimy całą informację o x_i. Dla x_i równoległego do v nie tracimy żadnej informacji. Tym samym łatwo jest się domyśleć, że najwięcej informacji zachowamy, jeśli zmaksymalizujemy $\cos^2(\angle(v, x_i))$. Tym samym należy zmaksymalizować $(v^T x_i)^2$. Chcemy jednak, by ta wielkość była zmaksymalizowana dla wszystkich x_i, więc zmaksymalizujmy

$$F(v) = \frac{1}{N}\sum_{i=1}^{N}(v^T x_i)^2 = \frac{1}{N}\sum_{i=1}^{N} v^T x_i v^T x_i =$$

$$= \frac{1}{N}\sum_{i=1}^{N} v^T x_i x_i^T v = v^T \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T\right) v = v^T C v$$

Warto tu się zatrzymać i zauważyć, że macierz kowariancji K danych podana wzorem

$$K = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

Dzięki naszemu wcześniejszemu wyśrodkowaniu danych sprowadza się do

$$K = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T = \frac{1}{N} X^T X = C$$

Więc otrzymana macierz C jest po prostu macierzą kowariancji zbioru danych określonego przez X i można ją wyznaczyć za pomocą wzoru $C = \frac{1}{N} X^T X$

Wracając do naszego problemu optymalizacyjnego, mamy funkcję
$F : \mathbb{R}^M \to \mathbb{R}$, dla której chcemy zmaksymalizować wartość $F(v)$ przy ograniczeniu $v^T v = 1$, co możemy zapisać jako $G(v) = v^T v - 1 = 0$

Możemy teraz skorzystać z metody mnożników Lagrange'a, gdzie w tym przypadku zmienię znak + na −, co mogę zrobić bez przeszkód i wtedy Lagrangian przyjmuje postać

$$\mathcal{L}(v, \lambda) = F(v) - \lambda G(v)$$

Dalej

$$\nabla_{v,\lambda}\mathcal{L}(v, \lambda) = \begin{cases} \nabla_v F(v) = \lambda \nabla_v G(v) \\ G(v) = 0 \end{cases}$$

$$\frac{\partial F}{\partial v_i} = \frac{\partial}{\partial v_i} v^T C v = \frac{\partial}{\partial v_i} \sum_{j=1}^{M} \sum_{k=1}^{M} v_j c_j^k v_k =$$

$$= \sum_{j=1}^{M} c_j^i v_i + \sum_{k=1}^{M} c_i^k v_k = \sum_{k=1}^{M} (c_k^i + c_i^k) v_k = \sum_{k=1}^{M} (c_i^k + c_i^k) v_k = 2 \sum_{k=1}^{M} c_i^k v_k = 2 \left[Cv \right]_i$$

$$\implies \nabla_v F(v) = 2Cv$$

Analogicznie

$$\nabla_v G(v) = 2v$$

$$\implies \nabla_v F(v) = -\lambda \nabla_v G(v) \iff 2Cv = 2\lambda v \iff Cv = \lambda v$$

Więc aby v spełniało rządane przez nas warunki, musi ono być wektorem własnym macierzy C. Zauważmy też, że

$$Cv = \lambda v \iff v^T C v = \lambda v^T v = \lambda$$

Więc w ekstremach funkcja $F(v)$ przyjmuje wartość równą λ. Tym samym, chcąc zmaksymalizować $F(v)$ należy wybrać największe λ

Kończąc nasze rozważania zauważmy, że $C = X^T X$, więc przypominając sobie rozkład SVD, w którym $X = U \Sigma V^T$, macierz C ma nieujemne wartości własne $\lambda_i = \sigma_i^2$, gdzie σ_i to kolejne wartości osobliwe macierzy X. Wtedy jeśli w SVD poukładaliśmy wartości malejąco na głównej przekątnej Σ, to

$$C = X^T X = V \Sigma^T \Sigma V^T$$

i kolumna v^i będzie wektorem jednostkowym własnym dla wartości λ_i

Chcąc uzyskać więcej wymiarów niż jeden, możemy chcieć dodać kolejną niezależną liniowo, najlepiej prostopadłą prostą, która byłaby niejako druga najbardziej znacząca. Na szczęście nie komplikuje nam to sprawy, jako że prosta ta spełniałaby w większości podobne założenia, a rozkład SVD w połączeniu z twierdzeniem spektralnym, z którego wynika, daje nam gotowe rozwiązanie - otóż gdy pierwsza prosta była generowana przez v^1, tak druga będzie generowana przez v^2, etc.

3 Wnioski

Jeśli zapiszemy

$$T = XV = U\Sigma$$

to macierz T będzie naszym zbiorem danych przetransformowanym tak, że kolejne kolumny reprezentują kolejne współrzędne wektorów po zmianie bazy na przez macierz V. Tym samym otrzymujemy hierarchiczną bazę przestrzeni, w której kolejne współrzędne są coraz mniej znaczące. Możemy teraz bezpiecznie

dokonać obcięcia macierzy V do V' tak, że V' to pierwsze kilka kolumn V. Wtedy $T' = XV'$ jest naszym optymalnie wyznaczonym zbiorem danych w przestrzeni niżej wymiarowej. Kolejne kolumny T są naszymi składowymi głównymi.

Tym samym algorytm wyznaczania składowych głównych stał się bardzo prosty

1. Wyśrodkuj macierz danych D poprzez odjęcie średniego wiersza $\bar{d}$ od każdego z wierszy macierzy D, otrzymaną macierz nazwij X
2. Dokonaj rozkładu SVD macierzy $X = U\Sigma V^T$
3. Wybierz tyle pierwszych kolumn macierzy V ile chcesz wymiarów i utwórz z nich macierz V'
4. XV' to twój przekształcony zestaw danych

4 Zadanka

W celu przedstawienia działania metody, oraz sprawdzenia kilku własności rozważmy prosty przykład. Niech D będzie macierzą o wymiarach 100×5, której kolumny oznaczają odpowiednio: wzrost, wagę, wiek, zarobki i powierzchnię mieszkania. W celu przedstawienia działania metody generowane dane zostały stworzone tak, aby wzrost był skorelowany z wagą, oraz zarobki z powierzchnią mieszkania.

$$D = \begin{pmatrix} 174 & 67 & 50 & 22768 & 282 \\ 195 & 85 & 62 & 14691 & 182 \\ 162 & 56 & 72 & 35906 & 447 \\ 163 & 58 & 62 & 12260 & 155 \\ 173 & 65 & 56 & 16806 & 208 \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

Następnie normalizujemy macierz, wyznaczamy średnie w kolejnych wierszach macierzy i odejmujemy je od nich. W ten sposób otrzymujemy w zaokrągleniu macierz odchyleń:

$$A = \begin{pmatrix} -0.1149 & 0.0250 & -0.0949 & -0.0340 & -0.0354 \\ 1.3297 & 1.4790 & 0.4980 & -0.7174 & -0.7400 \\ -0.9404 & -0.8635 & 0.9921 & 1.0776 & 1.1271 \\ -0.8716 & -0.7019 & 0.4980 & -0.9231 & -0.9303 \\ -0.1837 & -0.1365 & 0.2016 & -0.5385 & -0.5568 \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

Mając macierz odchyleń wyliczamy macierz kowariancji:

$$C = \begin{pmatrix} 0.9900 & 0.9832 & 0.0074 & -0.0448 & -0.0399 \\ 0.9832 & 0.9900 & 0.0120 & -0.0441 & -0.0392 \\ 0.0074 & 0.0120 & 0.9900 & -0.0404 & -0.0355 \\ -0.0448 & -0.0441 & -0.0404 & 0.9900 & 0.9891 \\ -0.0399 & -0.0392 & -0.0355 & 0.9891 & 0.9900 \end{pmatrix}$$

Jej wartości własne to: $[0.0008, 0.0068, 0.9870, 1.8930, 2.0624]$ i odpowiadające im unormowane wektory własne wynoszą:

$$V = \begin{pmatrix} -0.0086 & 0.7070 & -0.0090 & 0.5125 & -0.4871 \\ 0.0051 & -0.7071 & -0.0042 & 0.5128 & -0.4868 \\ -0.0035 & 0.0033 & 0.9985 & -0.0299 & -0.0451 \\ -0.7072 & -0.0066 & 0.0352 & 0.4852 & 0.5129 \\ 0.7069 & 0.0072 & 0.0401 & 0.4878 & 0.5105 \end{pmatrix}$$

Tylko dwie wartości własne są większe od jedynki. Ponieważ początkowo mieliśmy tylko 5 parametrów, to weźmiemy również wektor własny równy 0.9870, bo jest to wartość własna mniejsza od jedynki jej najbliższa. Zauważmy, że dla wektora własnego odpowiadającego największej wartości własnej parametry dochodu i powierzchni mieszkania są ze sobą skorelowane. Zarazem parametry wzrostu i wagi są przeciwnie do nich skorelowane. Dla drugiej największej wartości własnej cztery wspomniane parametry są skorelowane. W przypadku trzeciej wartości własnej widzimy, że parametr wieku nie jest skorelowany z żadnym elementem. Skoro pierwszy parametr jest skorelowany z drugim i czwarty z piątym ograniczenie przestrzeni do trzech wymiarów ma sens.
Przeprowadźmy rzutowanie danych na trójwymiarową przestrzeń.

$$D' = \begin{pmatrix} -0.0964 & -0.0770 & 0.0125 \\ 0.4241 & 0.7160 & -2.1359 \\ 1.0859 & 0.1183 & 1.9619 \\ 0.4383 & -1.7233 & -0.2046 \\ 0.1622 & -0.7031 & -0.4137 \end{pmatrix}$$

Jest to unormowany rzut macierzy D na przestrzeń trójwymiarową.

4.1 Uwagi

1. Istotnie wektory w macierzy rzutu są ortogonalne. Wynika to z faktu, że są to wektory własne odpowiadające innym wartościom własnym. Zatem muszą być względem siebie ortogonalne.

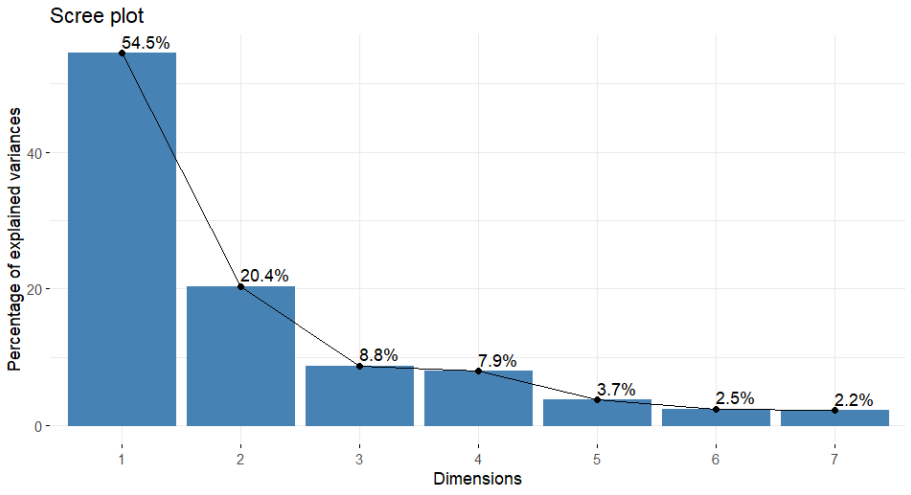
2. Innym sposobem na wyznaczenie wartości własnych do których chcemy zmniejszyć wymiar przestrzeni jest *Kryterium części wyjaśnionej wariancji*. Poczynając od największej wartości własnej wyznaczamy kolejno ich udział we wszystkich wartościach. Wybieramy te, których udział przekroczy łącznie 95%. W naszym przypadku trzy pierwsze wartości własne stanowią 99%.

5 Przykłady zastosowania metody PCA

5.1 Prezentacja rozmieszczania danych odzwierciedlających indeks szczęścia w różnych krajach z czynnikami społeczno - gospodarczymi

Dane, które będziemy rozważać zawierają wiersze reprezentujące kolejne państwa. Są one posortowane względem indeksu szczęścia w danym kraju. Zawierają kolejno kolumny: indeks szczęścia, dochód na mieszkańca, wsparcie społeczne, średnia oczekiwana długość życia w zdrowiu, wolność w życiowych wyborach, hojność obywateli, procent korupcji. Wszystkie te dane są ilościowe, ale najpierw musimy je unormować, tak by miały te same odchylenie standardowe. Następnie możemy wyznaczyć wartości własne. Oczywiście będzie ich tyle samo co kolumn. Przedstawia je poniższa tabela:

```
      eigenvalue variance.percent cumulative.variance.percent
Dim.1  3.8125442        54.464917                    54.46492
Dim.2  1.4271391        20.387702                    74.85262
Dim.3  0.6128853         8.755504                    83.60812
Dim.4  0.5563073         7.947247                    91.55537
Dim.5  0.2621029         3.744327                    95.29970
Dim.6  0.1723061         2.461516                    97.76121
Dim.7  0.1567151         2.238787                   100.00000
Ważność kolejnych składowych głównych i ich wpływ na wynik:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     1.9526 1.1946 0.78287 0.74586 0.51196 0.41510 0.39587
Proportion of Variance 0.5446 0.2039 0.08756 0.07947 0.03744 0.02462 0.02239
Cumulative Proportion  0.5446 0.7485 0.83608 0.91555 0.95300 0.97761 1.00000
```



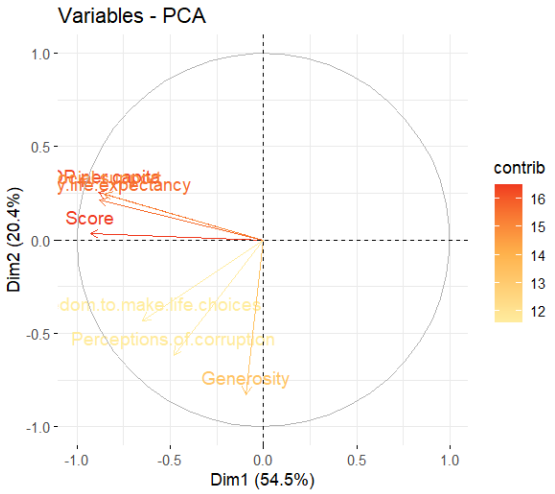
Widzimy, że dwa pierwsze wymiary tłumaczą łącznie ok. 75% wszytkich informacji w danych. Jest to dość sporo, możemy zatem je przedstawić na biplocie. Gdybyśmy chcieli uzyskać próg 95% tłumaczenia danych musielibyśmy wziąć 5 pierwszych składowych, czyli zredukowalibyśmy wymiar danych o 2.

Macierz ilości danej składowej głównej dla kolejnych rekordów w danych (kolejnych państw):

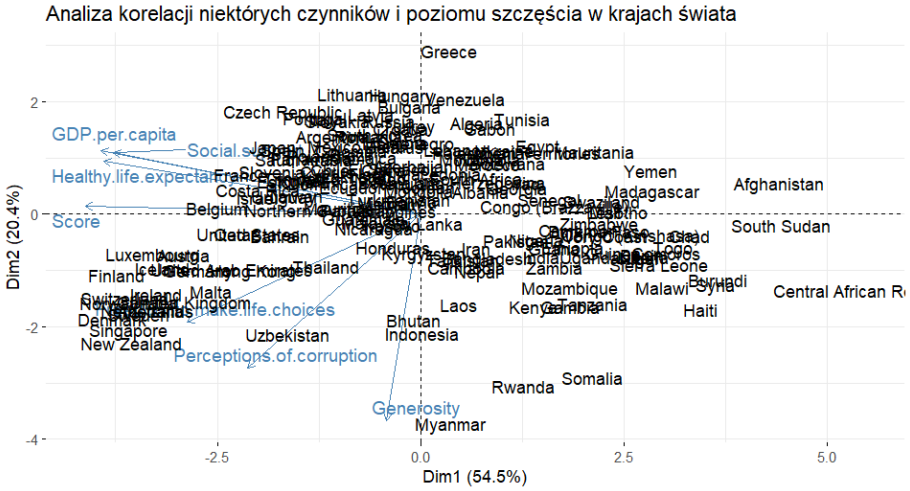
$$\begin{bmatrix}
-3.7343 & -1.0784 & -1.89372 & 0.3057 & 0.63299 & 0.47453 & -0.29398 \\
-3.7947 & -1.8656 & -1.46290 & -0.2285 & 0.57947 & 0.29340 & -0.14722 \\
-3.8220 & -1.5701 & -0.79071 & -0.2254 & 0.28594 & 0.23162 & 0.04927 \\
-3.1644 & -0.9587 & 1.54708 & -0.1829 & -0.08070 & 0.34207 & -0.00470 \\
-3.3558 & -1.7047 & -0.21467 & -0.5561 & 0.21647 & 0.37722 & -0.00749 \\
-3.6427 & -1.4862 & -0.92719 & -0.3854 & 0.16898 & 0.25118 & -0.06843 \\
-3.4574 & -1.7813 & -1.17730 & -0.3658 & 0.25852 & 0.23036 & -0.07792 \\
-3.5557 & -2.2915 & -0.77561 & -0.5503 & 0.42477 & 0.06808 & -0.26208 \\
-3.3492 & -1.5612 & -0.52321 & -0.2897 & 0.06681 & 0.16525 & -0.13404 \\
-2.9061 & -0.7166 & -0.17809 & -0.2397 & -0.06570 & 0.35056 & -0.05447 \\
-3.3077 & -1.7079 & -0.08059 & -0.6104 & 0.17296 & 0.10434 & -0.09724 \\
-1.9971 & 0.4249 & 0.31518 & 0.8527 & -0.22025 & 0.59221 & -0.57164 \\
-2.0015 & 0.2926 & 0.95417 & -0.9213 & -0.16010 & 0.58501 & -0.26319 \\
-3.2999 & -0.7088 & -1.24633 & -0.3515 & 0.05561 & 0.13273 & 0.36225 \\
-2.8284 & -1.5498 & 0.01905 & -1.1963 & 0.40312 & 0.15869 & -0.09774 \\
-3.2428 & -1.4268 & -0.50209 & -0.7621 & 0.31115 & 0.00373 & 0.21761 \\
-2.7313 & -1.0040 & -0.45163 & -0.5644 & 0.08735 & 0.21076 & 0.02382 \\
-2.5115 & 0.1104 & -0.59564 & -0.1205 & 0.13513 & 0.21672 & -0.06393 \\
-2.1129 & -0.3412 & 0.72634 & -0.5395 & 0.04696 & 0.43542 & 0.47736 \\
-1.6857 & 1.8233 & 0.09793 & 0.7338 & -0.10085 & 0.50214 & -0.05684 \\
-2.3236 & -0.9718 & 0.14829 & 0.2389 & -0.37677 & 0.42310 & 0.83593 \\
-2.5768 & -1.3771 & 1.25872 & -0.4337 & -0.16855 & -0.08028 & 0.03727 \\
-1.0548 & 1.2078 & -0.18308 & 0.5623 & -0.16410 & 0.59805 & -0.23589 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
2.7355 & -0.7574 & 0.66301 & 0.5638 & 0.08022 & -0.15755 & -0.63435 \\
3.0221 & -0.6945 & 0.21889 & -1.2153 & -0.05545 & 0.10685 & -0.43786
\end{bmatrix}$$

Obiekty były na początku posortowane, zatem różnice w korelacji między składowymi głównymi dla początkowych krajów są niewielkie, bo będą one rozmieszczone bliżej siebie na biplocie ze względu na podobieństwo względem większości cech. Wynika to też z tego, że pierwsza składowa jest skorelowana przeciwnie z kolumną odpowiadającą za indeks szczęścia. Na podstawie tego można łatwo wyciągnąć wniosek, że wizualizując dane państwa o wyższym indeksie szczęścia będą gęściej się pojawiać po lewej stronie wykresu. Z kolei widzimy, że ostatnie wartości macierzy w pierwszej kolumnie są dodatnie, więc państwa o najniższym indeksie szczęścia będą miały w większości dodatnie wartości w PCA1, dlatego będziemy ich się spodziewali po prawej stronie wykresu. Podobne rozważania możemy przeprowadzić dla PCA2, w tym przypadku rozważalibyśmy wartości

hojności obywateli, które tworzą największy cos kąta z drugą składową główną. Wszystko to dobrze obrazuje rozmieszczenie wektorów na płaszczyźnie:



Teraz pozostaje nanieść na wektory kolejne rekordy w danych, czyli w naszym przypadku nazwy państw. W ten sposób otrzymujemy końcowy rezultat w postaci biplotu, czyli wykresu, który obrazuje wszystko co mieliśmy w danych:



W tym przypadku biplot pomaga nam natychmiastowo określić korelacje pomiędzy niektórymi cechami obiektów, a także łatwo odnaleźć grupy państw o najsilniejszym natężeniu pewnej cechy. Minusem tego podejścia jest brak czytelności dla takich danych - przy ponad 100 unikatowych wierszach wyzwaniem może okazać się zlokalizowanie interesującego nas obiektu.

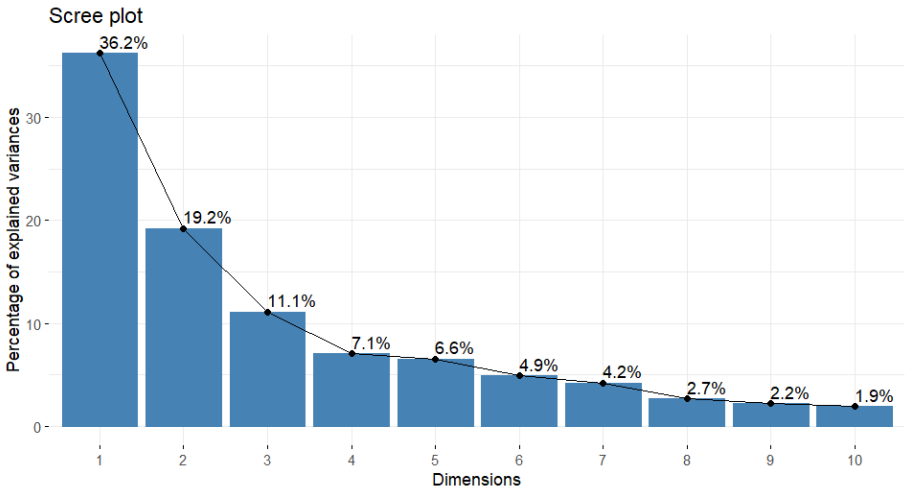
5.2 Wizualizacja danych pogrupowanych - przedstawienie trzech gatunków win i ich parametrów na jednym wykresie

Przyjrzyjmy się teraz danym, które są od razu podzielone w 3 grupy - różne szczepy win. Ramka danych zawiera 178 wierszy opisujących kolejne alkohole i 13 kolumn opisujących ilościowo ich parametry. Celem będzie sprawdzenie czy biplot dobrze odzwierciedli podobieństwo pomiędzy pewnymi winami oraz odszukanie próba scharakteryzowania tych szczepów. Zacznijmy tak jak w poprzednim przykładzie od wyznaczenia ważności kolejnych składowych głównych i stopnia w jakim wyjaśniają wariancję w danych:

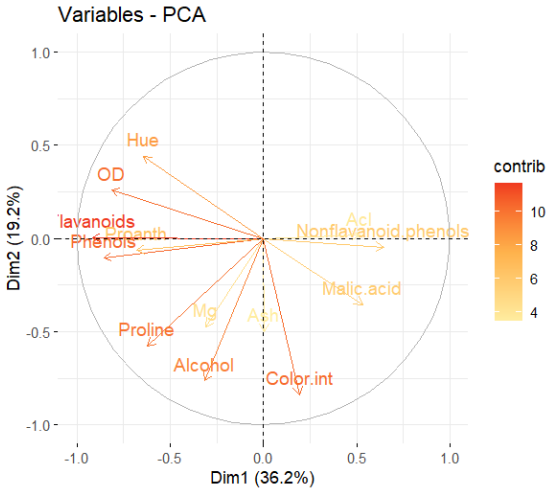
```
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
Standard deviation     2.169 1.5802 1.2025 0.95863 0.92370 0.80103 0.74231 0.59034
Proportion of Variance 0.362 0.1921 0.1112 0.07069 0.06563 0.04936 0.04239 0.02681
Cumulative Proportion  0.362 0.5541 0.6653 0.73599 0.80162 0.85098 0.89337 0.92018
                          PC9   PC10    PC11    PC12    PC13
Standard deviation     0.53748 0.5009 0.47517 0.41082 0.32152
Proportion of Variance 0.02222 0.0193 0.01737 0.01298 0.00795
Cumulative Proportion  0.94240 0.9617 0.97907 0.99205 1.00000
```

```
Poniżej przedstawione są kolejne wartości własne:
       eigenvalue variance.percent cumulative.variance.percent
Dim.1   4.7058503       36.1988481                    36.19885
Dim.2   2.4969737       19.2074903                    55.40634
Dim.3   1.4460720       11.1236305                    66.52997
Dim.4   0.9189739        7.0690302                    73.59900
Dim.5   0.8532282        6.5632937                    80.16229
Dim.6   0.6416570        4.9358233                    85.09812
Dim.7   0.5510283        4.2386793                    89.33680
Dim.8   0.3484974        2.6807489                    92.01754
Dim.9   0.2888799        2.2221534                    94.23970
Dim.10  0.2509025        1.9300191                    96.16972
Dim.11  0.2257886        1.7368357                    97.90655
Dim.12  0.1687702        1.2982326                    99.20479
Dim.13  0.1033779        0.7952149                   100.00000
```

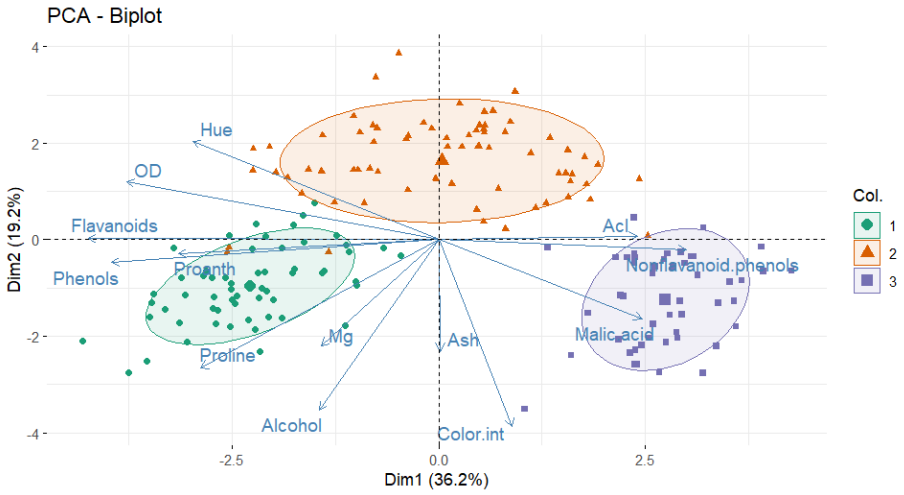
Latwo zauważyć, że w tym przypadku mamy zaledwie 55% dla dwóch wymiarów. Wynika to z faktu, że jest dwa razy więcej kolumn, więc biplot dwuwymiarowy jest słabszym przybliżeniem niż dla poprzedniego zestawu danych. Próg 95% jest tym razem osiągany dla dopiero 10 wymiaru.



Tak jak poprzednio przedstawmy wektory własne na płaszczyźnie:



Tu warto zauważyć, że cos kąta między wektorami odpowiadającymi za kierunek wzrostu czynnika odpowiadającego danej kolumni pokrywa się z korelacją tych wartości. Tu łatwo można zauważyć, że skorelowane są m.in: parametr Acl i NonFlavonoid Phenol. Przedstawmy teraz wszystkie alkohole na biplocie:



Końcowy rezultat daje nam bardzo jednoznaczny podział win w grupy. Elipsy, które obejmują najbliższe sobie 70% danych nie nachodzą na siebie, stąd widzimy rozdzielność w charakterystykach grup win. Możemy z tego wykresu wywnioskować, że wina z grupy 2 nie mają żadnego szczególnego czynnika (obszar elipsy jest największy i praktycznie kierunek żadnego wektora nie pokrywa się z jej wnętrzem, możemy jedynie zauważyć nieznaczną korelację ujemną z zawartością alkoholu, gdy przedłużymy wektor odpowiadający tej kolumnie). Z kolei dla grupy 1 łatwo dostrzec pewną korelację z ilością proliny i fenoli w składzie tych wina, a 3 charakteryzuje się sporą zawartością kwasu jabłkowego. Udało się zatem osiągnąć zamierzone rezultaty dzięki analizie składowych głównych.

Poniżej kod w języku R, który generuje powyższe wyniki i wykresy:

```r
dane <- read.csv("C:/Users/Admin/Downloads/2019.csv")
head(dane)
library(stats)
library(ggfortify)
library(ggplot2)
library(factoextra)
library(data.table)
library(dplyr)
dane<-data.table(dane)
dane[,c("Country.or.region")]
pca1 <- prcomp(dane %>% select(!(Overall.rank:Country.or.region)), scale=T)
fviz_eig(pca1,addlabels = T)
get_eig(pca1)
summary(pca1)
pca1$x
fviz_pca_ind(pca1,geom="point")
fviz_pca_var(pca1, col.var="contrib",gradient.cols="YlOrRd")
fviz_pca_biplot(pca1,geom="array",pch=20,habillage = dane$Country.or.region,repel=T) + geom_text(aes(label=dane$Country.or.region)) + labs(title = "Analiza korelacji niektórych czynników i poziomu szczęścia w krajach świata")
wina <- data.table(read.csv("C:/Users/Admin/Downloads/wine.csv"))
wina$Wine <- as.factor(wina$Wine)
pca2 <- prcomp(wina[,-c("Wine")],scale= T)
summary(pca2)
fviz_eig(pca2, addlabels = T)
get_eig(pca2)
pca2$x
fviz_pca_ind(pca2,geom="point",col.ind=wina$Wine)
fviz_pca_var(pca2, col.var="contrib",gradient.cols="YlOrRd")
fviz_pca_biplot(pca2,geom="point",col.ind=wina$Wine,addEllipses = T,ellipse.level=0.7,repel=T,palette="Dark2")
```

5.3 Analiza danych o wypadkach drogowych w Nowym Jorku wraz czynnikami pogodowymi

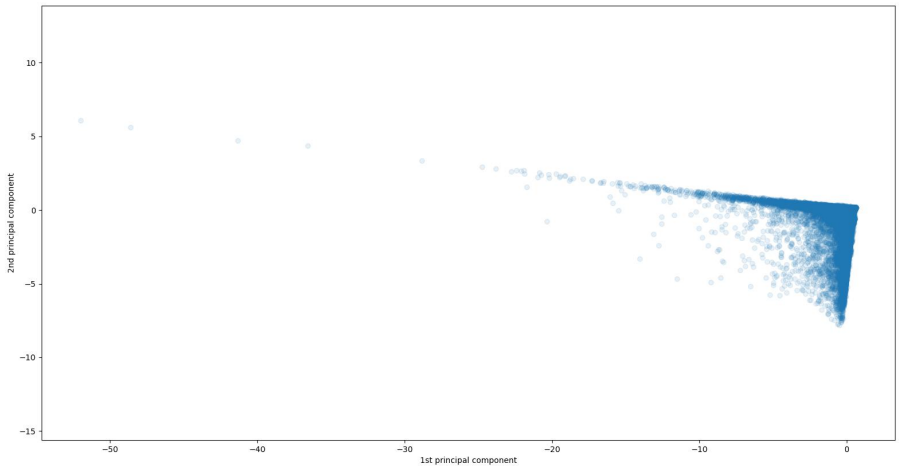
Rozważana dane są typu szeregu czasowego z częstotliwością godzinową, z wyjątkiem sytuacji braku danych dla jakieś godziny. Kolumny to: początkowe informacje o wypadkach drogowych - liczba wypadków, liczba osób poszkodowanych, liczba ofiar - a następne to parametry pogodowe - temperatura, deszcz, śnieg, opady (suma deszczu i śniegu) - oraz na końcu różne dane kategoryczne, nieistotne z perspektywy dalszej analizy. Dodatkowo w drugiej części, oznaczonej jako "Analiza 2", powyższe dane połączymy z danymi o opóźnieniach (w godzinach) autobusów szkolnych także w Nowym Jorku. Celem tych przykładów było przedstawienie analizy PCA bez użycia wyspecjalizowanych bibliotek, korzystając jedynie z pakietu scipy do rozkładu SVD.

```
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt


# Analiza nr 1

crash_weath = pd.read_csv("crash_weath.csv")
crash_weath = crash_weath.loc[crash_weath.loc[:,
↪  'precipitation'] > 0, ['nr_of_crashes', 'sum_injured',
↪  'temperature', 'cloud_cover', 'rain', 'precipitation']]

X= np.array(crash_weath)
n=X.shape[0]

# Próba normalizacji z użyciem średniej i odchylenia
↪  standardowego okazała się nieskuteczna,
# udział poszczególnych składowych wyglądał następująco:
↪  [0.31256521 0.48960962 0.63813753 0.77827171 0.8960387
↪  0.99999979  1.        ]
# Dane mają silnie prawoskośny charakter, więc zastąpimy średnią
↪  medianą, a odchylenie standardowe IQR (rozstępem
↪  międzykwartylowym)
Z = (X - np.median(X, axis=0))/(np.quantile(X, 0.75,
↪  axis=0)-np.quantile(X, 0.25, axis=0))

U, s, V = scipy.linalg.svd(Z, full_matrices=False)
#Sprawdzamy udział poszczególnych składowych
print(np.cumsum(s**2)/np.sum(s**2))
# Output
```

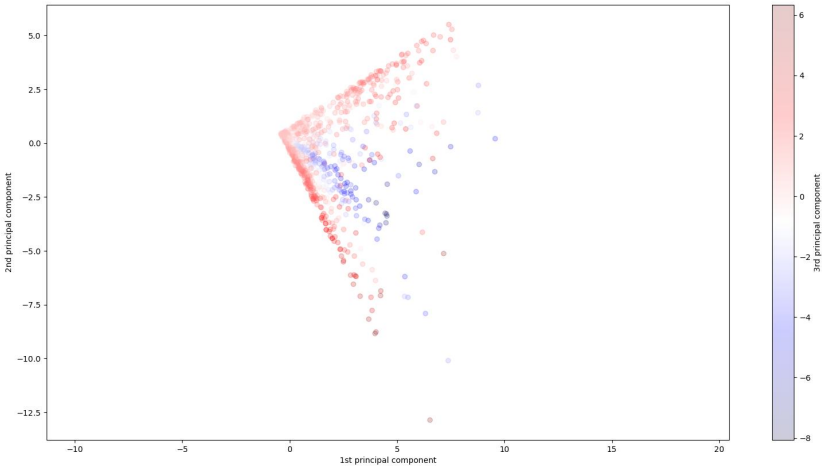
```
# [0.5213516  0.85070863 0.94689292 0.97587945 0.9945049  1.
↪  ]
# Mimo, że dwie składowe odpowiadają tylko za 85%, dodanie
↪  trzeciej nie wpływa na wygląd wykresu, dlatego ją pominiemy
print(np.round(V[0, :], 2))     #Output [-0.04 -0.05 -0.03  0.11
↪  -0.69 -0.71]
print(np.round(V[1, :], 2))     #Output [-0.08 -0.09 -0.12  0.98
↪  0.08  0.08]


# Tworzymy wykres
P2 = U[:, :2] @ np.diag(s[:2])
plt.plot(P2[:, 0], P2[:, 1], "o", alpha=0.1)
plt.axis("equal")
plt.xlabel("1st principal component (52%)")
plt.ylabel("2nd principal component (33%)")
plt.show()


# Analiza nr 2

bcw = pd.read_csv("bus_crash_weath.csv")
bcw = bcw.loc[bcw.loc[:, 'precipitation'] > 0, ['nr_of_crashes',
↪  'temperature', 'cloud_cover', 'rain', 'precipitation',
↪  'delay_hr']]

X= np.array(bcw)
n=X.shape[0]

# Dane mają silnie prawoskośny charakter, więc zastąpimy średnią
↪  medianą, a odchylenie standardowe IQR (rozstępem
↪  międzykwartylowym)
Z = (X - np.median(X, axis=0))/(np.quantile(X, 0.75,
↪  axis=0)-np.quantile(X, 0.25, axis=0))
U, s, V = scipy.linalg.svd(Z, full_matrices=False)
#Sprawdzamy udział poszczególnych składowych
print(np.cumsum(s**2)/np.sum(s**2))
# Output
# [0.40882078 0.72066429 0.93498726 0.97232638 0.99823042 1.
↪  ]  - wybieramy 3 składowe
print(np.round(V[0, :], 2))     #Output [ 0.04  0.08 -0.8   0.24
↪  0.26  0.47]
print(np.round(V[1, :], 2))     #Output [ 0.04  0.04 -0.53 -0.53
↪  -0.57 -0.33]
print(np.round(V[2, :], 2))     #Output [-0.02  0.08 -0.24  0.36
↪  0.37 -0.82]
```

```
57
58
59  # Tworzymy wykres
60  P2 = U[:, :3] @ np.diag(s[:3])
61  plt.scatter(
62      P2[:, 0],
63      P2[:, 1],
64      c=P2[:, 2],
65      cmap=plt.colormaps.get_cmap("seismic"),
66      alpha=0.2
67  )
68  plt.xlabel("1st principal component (41%)")
69  plt.ylabel("2nd principal component (31%)")
70  plt.axis("equal")
71  plt.rcParams["axes.grid"] = False
72  cbar = plt.colorbar()
73  plt.rcParams["axes.grid"] = True
74  cbar.set_label('3rd principal component (22%)')
75
76  plt.show()
```

Poniższy wykres przedstawia dwie główne składowe otrzymane w "Analizie 1". Wybranie również trzeciej składowej nie wywoływało widocznych zmian na wykresie, chociaż pierwsze dwie tłumaczą ledwie 85%, dlatego zdecydowaliśmy się z niej zrezygnować. Otrzymany wykres można interpretować jako potwierdzenie powszechnie znanej obserwacji meteorologicznej, a mianowicie: intensywne opady są powiązane z dużym zachmurzeniem, natomiast przy niewielkich opadach niebo niekoniecznie będzie w znacznym stopniu zachmurzone.



Niektóre wykresy potrafią przyjąć także mniej oczywisty układ niż ten powyższy. Przykładowo na wykresie poniższej, otrzymanym w "Analizie 2", widzimy, że korelacje między zmiennymi reprezentującymi opóźnienie, opady i zachmurzenie nie są oczywiste i interpretacja ich bez użycia innych, zaawansowanych narzędzi statystycznych może rodzić duże trudności.



5.4 Analiza udziału przychodów z zasobów naturalnych w PKB

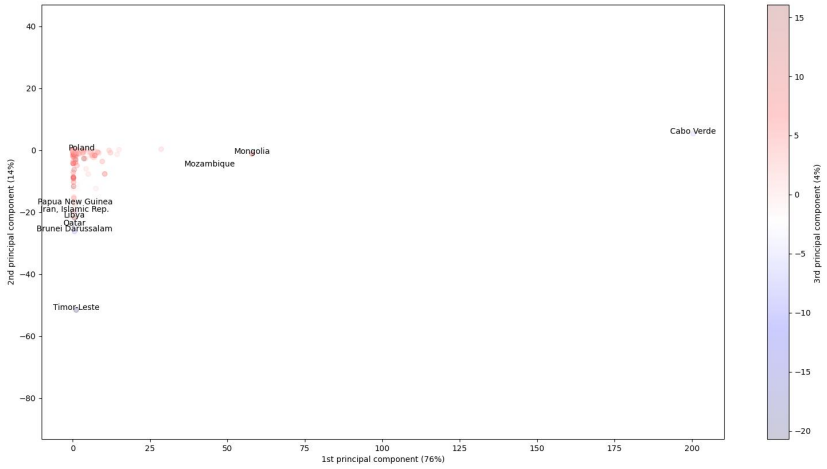
Teraz zajmiemy się danymi, których wiersze reprezentują kolejne kraje lub regiony geograficzno-kulturowe. Natomiast kolumny to: procentowy udział przychodów ze wszystkich surowców naturalnych w PKB danego państwa/wspólnoty gospodarczej w 2021 roku, a następnie procentowy udział tych przychodów w PKB kolejno dla ropy, gazu, węgla, kruszców (na przykład złota) i drewna. Konieczne było sformatowanie danych do powyżej opisanej postaci w celu przeprowadzenia na nich analizy. Podobnie jak w poprzednich przykładach dane nie cechowały się rozkładem normalnym, lecz prawoskośnym, dlatego do procesu normalizacji użyto medianę i IQR (rozstęp międzykwartylowy).

```
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt

# Analiza nr 3

data=pd.read_csv("contribution_of_natural_resources_to_GDP.csv")

# Formatowanie danych tak, aby każdy wiersz reprezentował jeden
↪  kraj/region, a kolumna kolejne parametry
data['Total natural resources rents (% of GDP)']=
↪  data.iloc[:266,2]
data['Oil rents (% of GDP)']= 0
data.loc[:,'Oil rents (% of GDP)'].iloc[:266] =
↪  data.iloc[266:2*266,2]
data['Natural gas rents (% of GDP)'] =0
data.loc[:,'Natural gas rents (% of GDP)'].iloc[:266]=
↪  data.iloc[2*266:3*266,2]
data['Coal rents (% of GDP)']= 0
data.loc[:,'Coal rents (% of GDP)'].iloc[:266]=
↪  data.iloc[3*266:4*266,2]
data['Mineral rents (% of GDP)']= 0
data.loc[:,'Mineral rents (% of GDP)'].iloc[:266]=
↪  data.iloc[4*266:5*266,2]
data['Forest rents (% of GDP)']= 0
data.loc[:,'Forest rents (% of GDP)'].iloc[:266]=
↪  data.iloc[5*266:6*266,2]
formatted_data= data.iloc[:266,].drop(['Series
↪  Name','2021'],axis=1).dropna().reset_index(drop=True)


X= np.array(formatted_data.iloc[:,1:])
```

```
n=X.shape[0]

# Dane mają prawoskośny charakter, więc zastąpimy średnią medianą, a odchylenie standardowe IQR (rozstępem międzykwartylowym)
Z = (X - np.median(X, axis=0))/(np.quantile(X, 0.75, axis=0)-np.quantile(X, 0.25, axis=0))

U, s, V = scipy.linalg.svd(Z, full_matrices=False)
#Sprawdzamy udział poszczególnych składowych
print(np.cumsum(s**2)/np.sum(s**2))
# Output
# [0.75751005 0.89546106 0.94137571 0.98286056 0.99986687 1. ]   - wybieramy 3 składowe
print(np.round(V[0, :], 2))     #Output  [0.02 0.01 0.02 1. 0.04 0.01]
print(np.round(V[1, :], 2))     #Output  [-0.13 -0.43 -0.88 0.03 -0.1  -0.03]
print(np.round(V[2, :], 2))     #Output  [ 0.19  0.61 -0.41 -0.03  0.6   0.27]


# Tworzymy wykres
P2 = U[:, :3] @ np.diag(s[:3])
plt.scatter(
    P2[:, 0],
    P2[:, 1],
    c=P2[:, 2],
    cmap=plt.colormaps.get_cmap("seismic"),
    alpha=0.2
)
# Wybieramy wyróżniające się kraje
inds= formatted_data[formatted_data['Country Name'].isin(['Poland','Cabo Verde','Timor-Leste', 'Brunei Darussalam',
    'Papua New Guinea','Mongolia','Mozambique','Iran, Islamic Rep.','Qatar','Libya'])].index.to_numpy()

# Dodajemy etykiety wyróżniających się krajów
for i in inds:
    plt.text(P2[i, 0], P2[i, 1], formatted_data.iloc[i,0], ha="center")

plt.axis("equal")
plt.xlabel("1st principal component (76%)")
plt.ylabel("2nd principal component (14%)")
```

```
plt.rcParams["axes.grid"] = False
cbar = plt.colorbar()
plt.rcParams["axes.grid"] = True
cbar.set_label('3rd principal component (4%) ')
plt.show()

```

Na wykresie dwóm pierwszym składowym odpowiadają osie X i Y, natomiast trzecia składowa jest zaprezentowana w formie skali kolorów. Dodanie etykiet dla wszystkich państw spowodowałoby nałożenie się ich i w efekcie brak czytelności, dlatego postanowiliśmy ręcznie wybrać etykiety dla wyróżniających się państw. Z wykresu można odczytać, że większość krajów, w tym wszystkie uważane za średnio i wysoko rozwinięte, na przykład Polska, nie opiera swojej gospodarki na zyskach z surowców naturalnych. Są natomiast państwa, w których rozkład PKB jest zupełnie inny, na przykład Katar i Timor Wschodni w przypadku ropy, czy Mozambik w przypadku węgla.



6 Zalety i wady użycia analizy składowych głównych

Zalety:

1. Zmniejszenie wymiaru danych - może to przyspieszać trenowanie modeli uczących się.
2. Skuteczna wizualizacja złożonych danych na płaszczyźnie.

3. Pomaga pokazać różnice i podobieństwa między pewnymi grupami danych.

Wady:

1. Szybkość działania PCA rośnie sześciennie z wymiarem danych - algorytm może okazać się czasochłonny dla większych danych.
2. Redukcja szumu w danych i pozbawienie pewnych wymiarów prowadzi do utraty informacji, a to może być czasem niepożądane.
3. Wartości odstające mogą łatwo zaburzyć końcowy wynik.
4. W niektórych sytuacjach może pojawić się trudność w interpretacji znaczenia składowych, które są bardziej złożone niż zwykłe zmienne.