Лабораторна робота №1
Студента 2 курсу магістратури
Групи "статистика"
Варіант №4

Горбунов Даніел Денисович

19 жовтня 2022 р.

Частина перша.

Вступ.

У даній роботі побудовано оптимальну кластеризацію на даних з текстового файлу "mult4.txt". Висновки про якість розбиття були зроблені на основі відповідних метрик (внутрішньогрупова сума квадратів, середні силуети).

Хід роботи.

Підготовча робота над даними.

Першочергово треба розібратися з тим, що за дані записані у файлі.

```
> # Зчитуємо дані
> data <- read.table("./mult4.txt", header=T)
> # Розмірність таблиці
> c(nrow(data), ncol(data))
```

```
[1] 1000   50
```

Працюємо з таблицею з 1000 об'єктів, кожному відповідє числовий вектор з 50 характеристик. Що ці характеристики представляють з себе – попередньо про це ніде не сказано. Подивимося на середні та стандартні відхилення характеристик:

```
> # Підрахунок середніх за кожною характеристикою
> round(apply(data, 2, mean), 2)
```

```
   V1    V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12   V13
 0.11 -2.35  1.18  4.77  3.88 -0.90  1.67  0.65  2.24 -0.33  0.30 -1.63 -0.22
  V14   V15   V16   V17   V18   V19   V20   V21   V22   V23   V24   V25   V26
 1.01 -2.54  1.77  1.08  2.15 -0.35 -0.06  0.45  2.17  3.10  0.24 -3.06 -2.44
  V27   V28   V29   V30   V31   V32   V33   V34   V35   V36   V37   V38   V39
 3.32 -0.57 -1.88 -2.07  1.28  1.03 -1.19  1.26 -1.06  0.85  0.60  1.78 -0.35
  V40   V41   V42   V43   V44   V45   V46   V47   V48   V49   V50
 2.53  0.49  1.98 -0.24  0.54  2.26  1.23  0.04 -0.54 -0.66 -0.07
```

```
> # Підрахунок коренів з дисперсії за кожною характеристикою
> round(apply(data, 2, sd), 2)

  V1   V2   V3   V4   V5   V6   V7   V8   V9  V10  V11  V12  V13  V14  V15  V16
4.63 1.42 1.84 3.22 3.32 2.10 1.45 3.93 1.89 2.49 2.46 0.95 2.46 2.64 1.74 2.23
 V17  V18  V19  V20  V21  V22  V23  V24  V25  V26  V27  V28  V29  V30  V31  V32
1.35 2.62 2.91 1.51 0.79 1.23 0.86 2.60 1.50 1.18 4.00 1.36 1.38 2.91 3.45 2.26
 V33  V34  V35  V36  V37  V38  V39  V40  V41  V42  V43  V44  V45  V46  V47  V48
3.44 1.66 2.01 3.98 1.42 1.08 4.13 1.51 4.29 1.44 2.23 2.38 0.93 2.19 3.32 2.05
 V49  V50
1.48 1.76
```

Судячи з первинних значень про центральне положення та розкиду, то наврядчи можна вважати розподіл характеристик в певній мірі однорідним. Тому надалі стандартизуємо значення кожної характеристики в таблиці:

```
> # Стандартизуємо дані
> data.std <- scale(data)
```

Тепер можна переходити до основної частини роботи.

Метод центроїдів.

Застосуємо метод центроїдів на стандартизованій таблиці. Але для того, щоб його застосувати, треба вказати скільки кластерів треба утворити. Оскільки наперед ми не знаємо яку кількість розбиттів доречно задати в алгоритмі, спочатку визначимо кандидатів на основі внутрішньогрупового розкиду та силуетів. Перебирати будемо в рамках від 1 до 20 кластерів.

```
> # Кластеризація, 1 - 20 штук
> K.max <- 20
```

Почнемо з "оптимізації" на основі внутрішньогрупової дисперсії. Обчислення реалізуємо використовуючи функцію *fviz_ nbclust* з пакету *factoextra*.

```
> # Вибір кількості кластерів на основі значення WSS
> fviz_nbclust(
+   x = data.std,
+   FUNcluster = kmeans,
+   method = "wss",
+   k.max = K.max
+ )
```

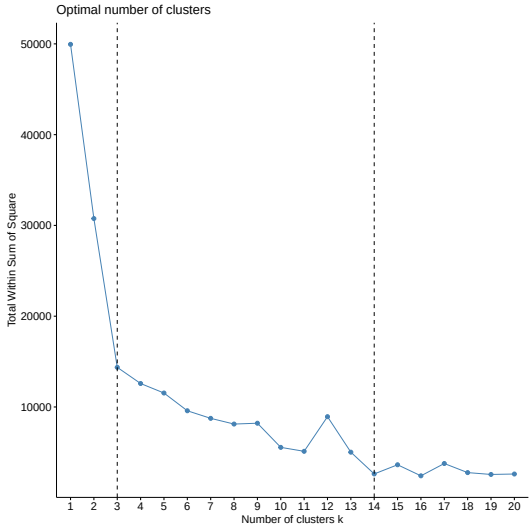

Рис. 1: Внутрішньогрупові суми квадратів. Кластеризація методом центроїдів.

На рисунку видно злам при $k = 3$ кластерах. З іншого боку, ефект затухання спостерігається до певного моменту, а саме до $k = 12$, де спостерігається підйом міжгрупової дисперсії вгору. Після цього значення спадають, тому на роль ще одного кандидата можна взяти $k = 14$ (бо найменша кількість кластерів з найменших розкидом). Подивимося на те, що вимальовується на середніх силуетах.

```
> # Вибір кількості кластерів на основі середніх силуетів
> fviz_nbclust(
+   x = data.std,
+   FUNcluster = kmeans,
+   method = "silhouette",
+   k.max = K.max
+ )
```

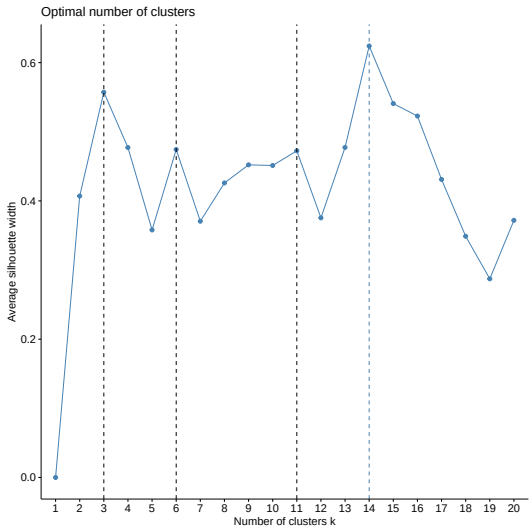

Рис. 2: Діаграма середніх силуетів. Кластеризація методом центроїдів.

На діаграмі силуетів видно, що на $k \in \{3, 6, 11, 14\}$ кластерах, в середньому, точки розміщуються вдало. Отже для подальшого дослідження, доведеться розібратися з якістю кластеризації на k частин, де $k \in \{3, 6, 11, 14\}$.

Кластеризація при $k = 3$. Зафіксуємо деяку зернину для можливого відтворення результатів в майбутньому та проведемо кластеризацію методом центроїодів:

```
> # Працюємо з кластеризацією з 3-х кластерів
> set.seed(777)
> data.kmeans <- kmeans(data.std, 3, nstart = 50)
> # Виводимо "звіт" по результатам процедури
> data.kmeans[ c("totss", "withinss", "tot.withinss", "betweenss")]

$totss
[1] 49950

$withinss
[1] 4293.064 3931.980 6143.270

$tot.withinss
[1] 14368.31

$betweenss
[1] 35581.69
```

На жаль, ми ще не навчилися дивитися на рисунки у просторах розмірності вище трьох, але можна зменшити розмірність. Для початку застосуємо метод головних компонент на стандартизованих даних, де далі зробимо проекцію на перші дві компоненти.

```
> # Метод головних компонент на стандартизованих даних
> data.pca <- princomp(data.std, cor = F)
> plot(data.pca, main="Діаграма власних чисел, PCA")
```

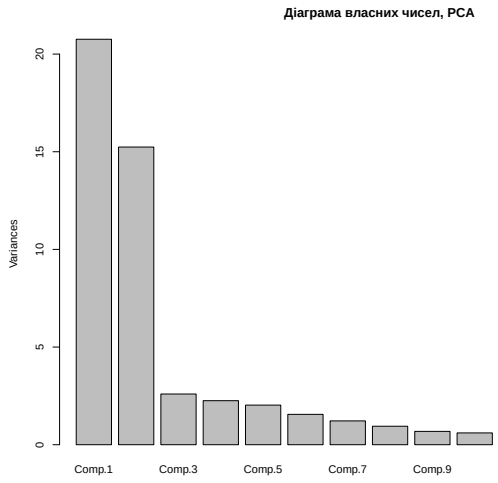

Рис. 3: Діаграма власних чисел у методі головних компонент.

Зауважимо, що на перші дві компоненти припадає приблизно 72% від загальної дисперсії (точніше, то на першу ≈41%, а на другу ≈31%).

Безпосередньо візуалізація:

```
> # Проекція на перші дві компоненти, візуалізація
> pal <- c("red", "green", "blue")
> plot(data.pca$scores[,1:2], col=pal[data.kmeans$cluster], cex=0.5)
```

В результаті можна побачити таке:

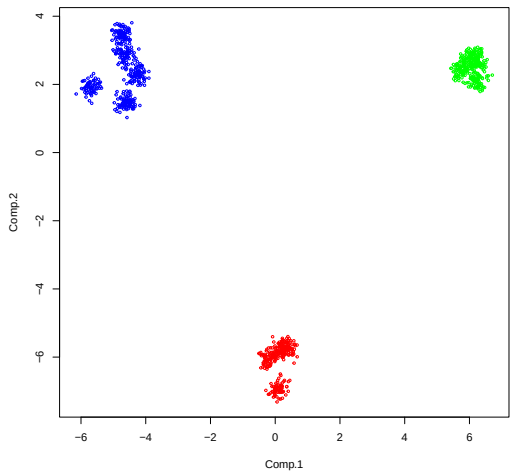

Рис. 4: Діаграма розсіювання даних на перші дві головні компоненти.

Дійсно, на проекції добре видно розмежування даних на три великих кластери. Хоча є нюанс: у той час коли до "зеленого" кластера питань не виникає (насправді це не так, питання то буде), то для "синього" та "червоного" можна побачити наявність дрібніших компактних кластерів. Тому ідея робити кластеризацію на $k = 3$ частини здається коректною для вхоплення загальних властивостей.

Питання з "зеленим" досить делікатне. Якщо придивитися, то цей кластер теж складається з підкластерів, але які розміщені близько. Чи потрібно це враховувати (заморочуватися над тим, аби врахувати це зауваження), або ні – розберемося далі (хоча ні те, ні інше не фактичної інформації дає вдосталь).

Побудуємо діаграму розсіювання даних зі зменшенням розмірності на основі методу канонічних компонент.

```
> # Отримаємо номери кластерів, до яких відносяться об'єкти
> cluster.idx <- data.kmeans$cluster
> # Кількість кластерів
> clust.num <- length(levels(as.factor(cluster.idx)))
> # Кількість об'єктів
> n <- nrow(data.std)
> # Застосування CCA для таблиці
> C <- matrix(
+   data = as.numeric(rep(cluster.idx, clust.num) == rep(1:clust.num, each = n)),
+   ncol = clust.num,
+   nrow = n)
> cc_res<-rcc(data.std, C, 0.1, 0.1)
> plot(
+   cc_res$scores$xscores[,1:2],
+   col=pal[cluster.idx],
+   cex=0.5
+   )
```

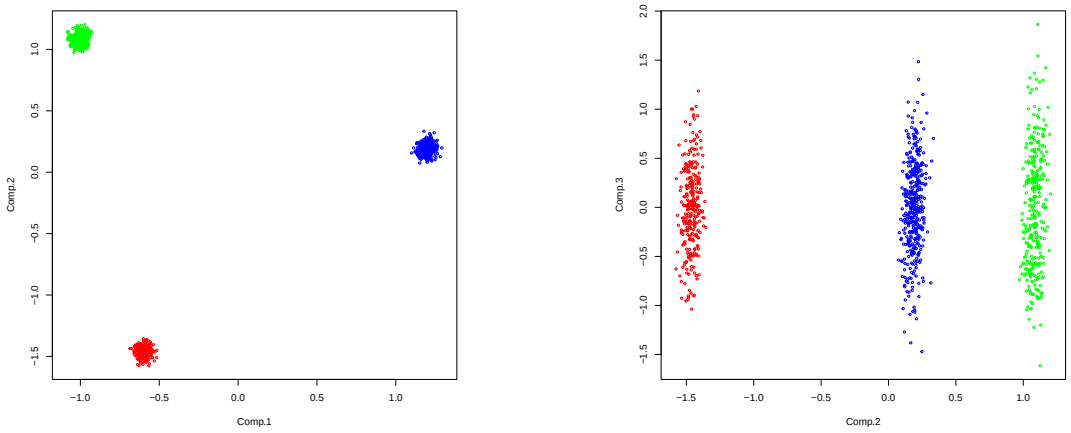

Рис. 5: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

По суті картина не сильно змінилася, як у випадку використання методу головних компонент. Варто зазначити, що у порівнянні з попереднім методом ми не спостерігаємо наявність підгруп кластерів на рисунках.

Усі кроки для наступних випадків аналогічні попереднім, тому надалі коментарі будуть лише у разі необхідності (інтерпретація результатів).

Кластеризація при $k = 6$.

```
> # Працюємо з кластеризацією з 6-и кластерів
> set.seed(777)
> data.kmeans <- kmeans(data.std, 6, nstart = 50)
> # Виводимо "звіт" по результатам процедури
> data.kmeans[ c("totss", "withinss", "tot.withinss", "betweenss")]

$totss
[1] 49950

$withinss
[1] 1488.5975 1256.1322  128.1864 3931.9796 1090.3874 1252.4955

$tot.withinss
[1] 9147.779

$betweenss
[1] 40802.22
```

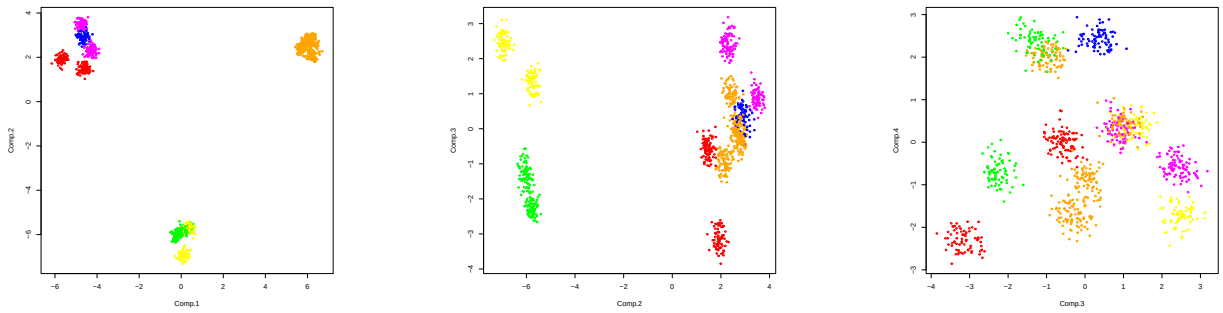

Рис. 6: Діаграма розсіювання даних на перші чотири головні компоненти.

Не зовсім зрозуміла природа того, як віднесли відповідні хмаринки даних до "рожевого" класу. Треба подивитися на це з іншої перспективи (змінити вісі проектування головних компонент, що і зробили). На відповідних проекціях не видно, аби відповідні хмарини утворювали клстери – деякі з них взагалі змішуються. Спробуємо розібратися за допомогою проектування на канонічні компоненти.

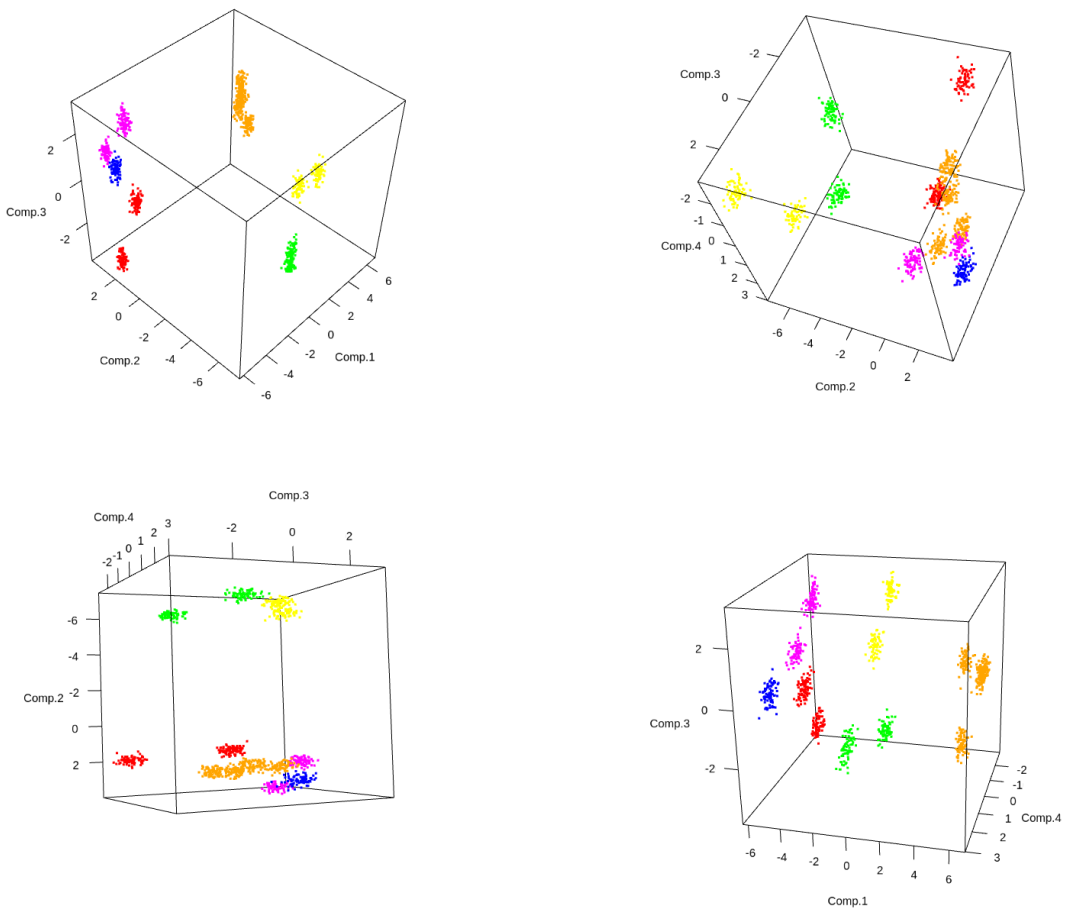

Рис. 7: Просторова діаграма розсіювання на перші чотири головні компоненти.

Виявилося, що вихід у простір дав трохи розуміння того, чи має місце запропоноване розбиття чи ні. Справді, покрутивши трохи на різних проекціях діаграми, можна побачити, що виділені кластери відокремлюються. З іншого боку, наприклад якщо взяти проекцію на вісі, що відповідають першій, третій та четвертій головним компонентам, "червона" кластеризація виглядає сумнівною з цієї перспективи: одна з хмаринок лежить ближче до хмаринок з інших кластерів, ніж до власної.

Побудуємо діаграму розсіювання даних зі зменшенням розмірності на основі методу канонічних компонент.

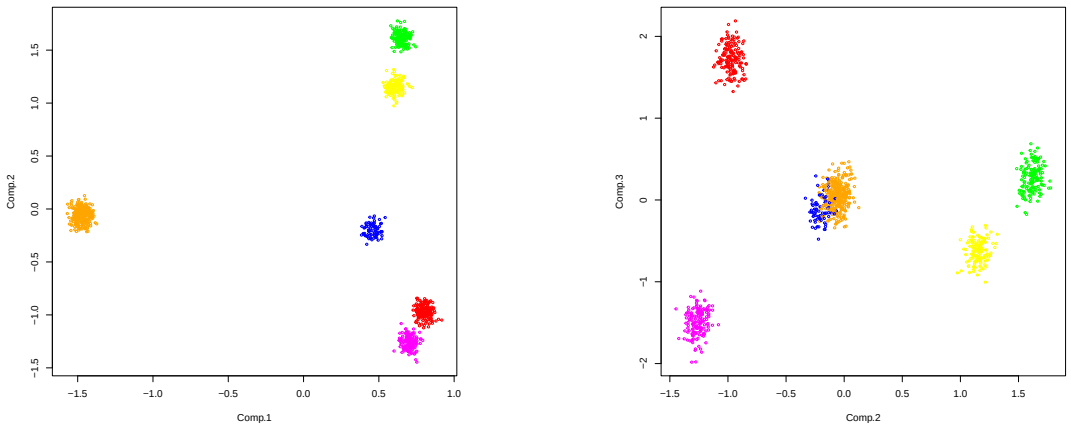

Рис. 8: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

Тут утворені кластери відділяються один від одного на одній вісі, хоча на іншій проекції видно змішування обох кластерів (”синій” та ”помаранчевий”). Вийдемо з площини на простір:

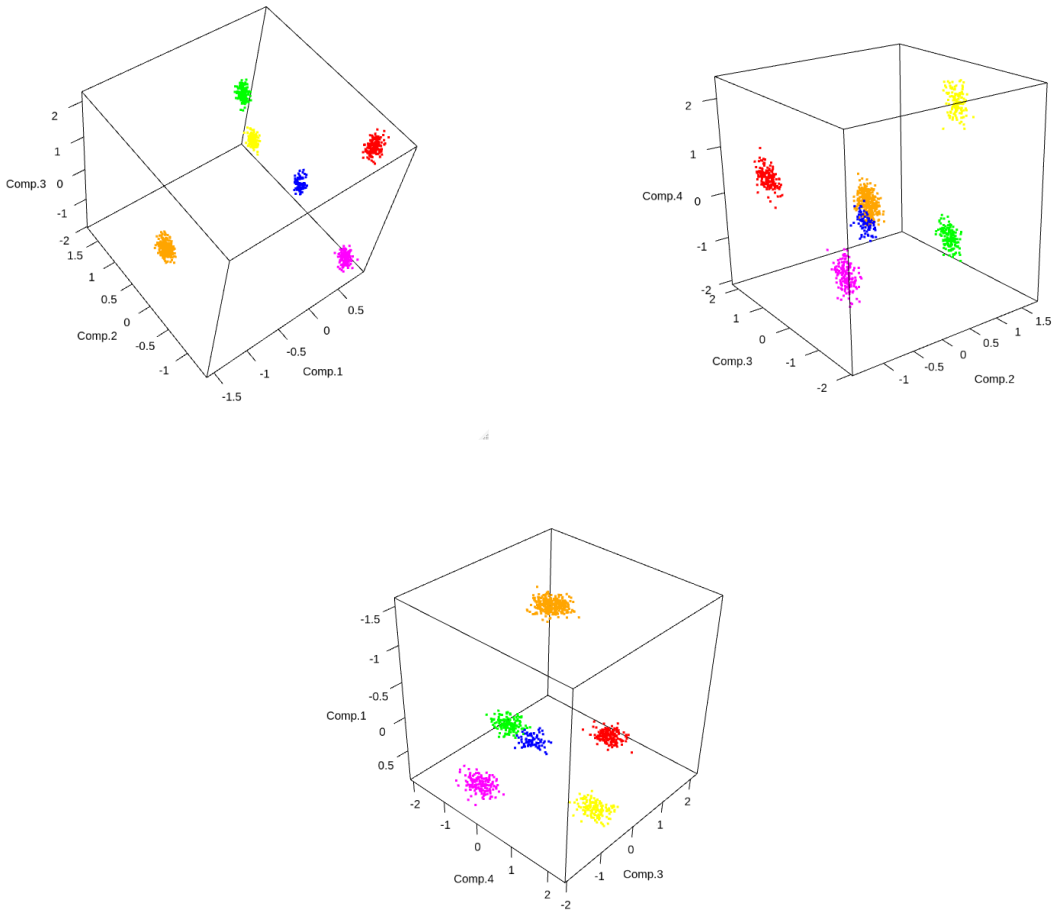

Рис. 9: Просторова діаграма розсіювання на перші чотири канонічні компоненти.

Розділення видно. А на деякій з проекцій можна міркувати аналогічно як на основі діаграми розсіювання на площині.

Кластеризація при $k = 11$.

```
> # Працюємо з кластеризацією з 11-и кластерів
> set.seed(777)
> data.kmeans <- kmeans(data.std, 11, nstart = 50)
> # Виводимо "звіт" по результатам процедури
> data.kmeans[ c("totss", "withinss", "tot.withinss", "betweenss")]

$totss
[1] 49950

$withinss
 [1]  128.1864  118.6839  108.2549  116.8799 1252.4955  155.7802  130.9681
 [8]  114.0970  140.8986  138.0156 1256.1322

$tot.withinss
[1] 3660.392

$betweenss
[1] 46289.61
```

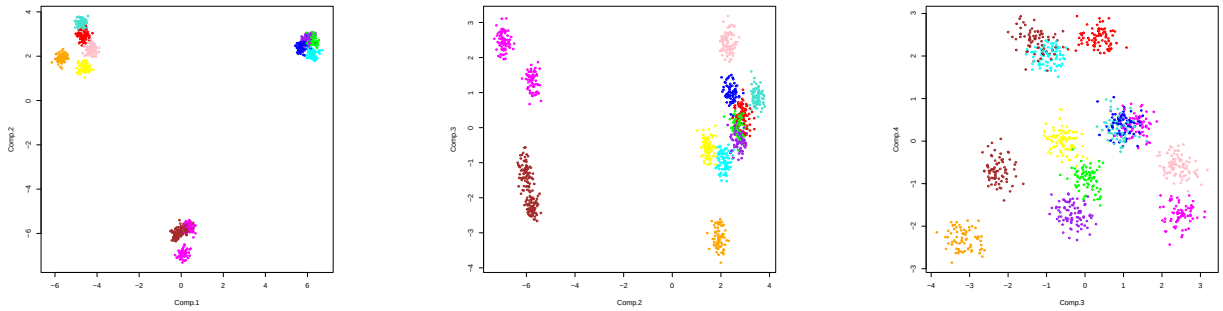

Рис. 10: Діаграма розсіювання даних на перші чотири головні компоненти.

Сумбурно все це виглядає. Можливо, на тривимірних проекціях буде зрозуміліше?

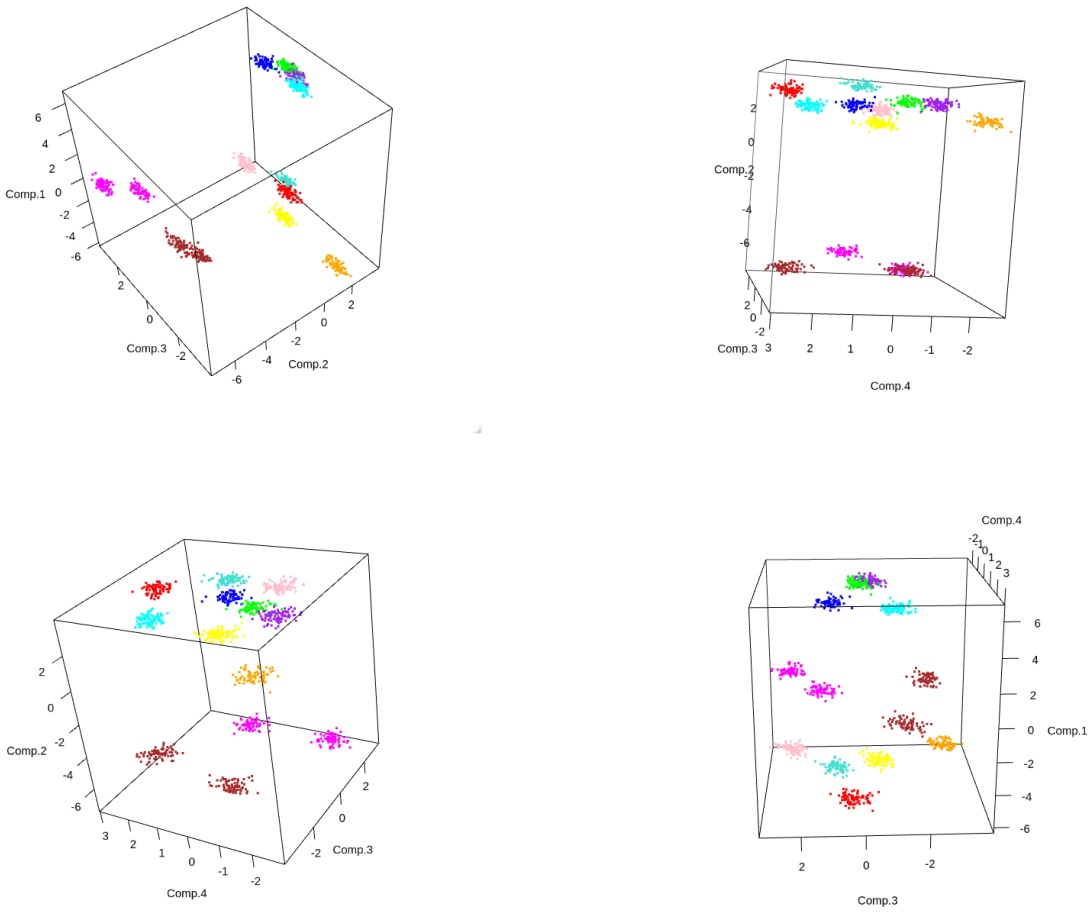

Рис. 11: Просторова діаграма розсіювання на перші чотири головні компоненти.

На проекціях на перші вісі головних компонент виходить така картина, що кластеризація наче вийшла порівняно краще ніж у попередньому випадку. Подивившися з різних точок на діаграму, розмітка вигладає нормально. Також зауважимо близкість (змішування) двох кластерів: ”зеленого” та ”фіолетового”.

Побудуємо діаграму розсіювання даних зі зменшенням розмірності на основі методу канонічних компонент.

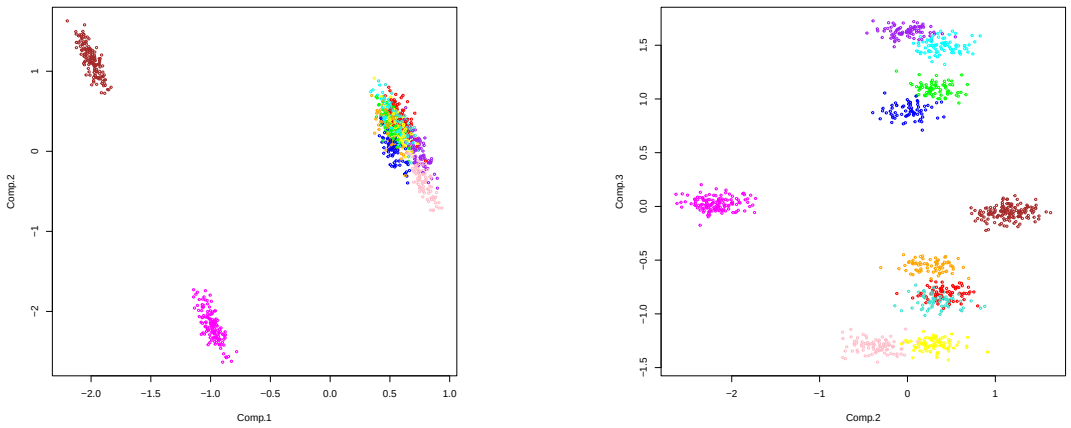

Рис. 12: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

На одній з осей видно, що багато кластерів скупчується в один. Можливо просто не з невдалої сторони подивилися, тому така дивна картинка вийшла. Це якраз мотивує подивитися на просторові проекції, чим займемося далі.

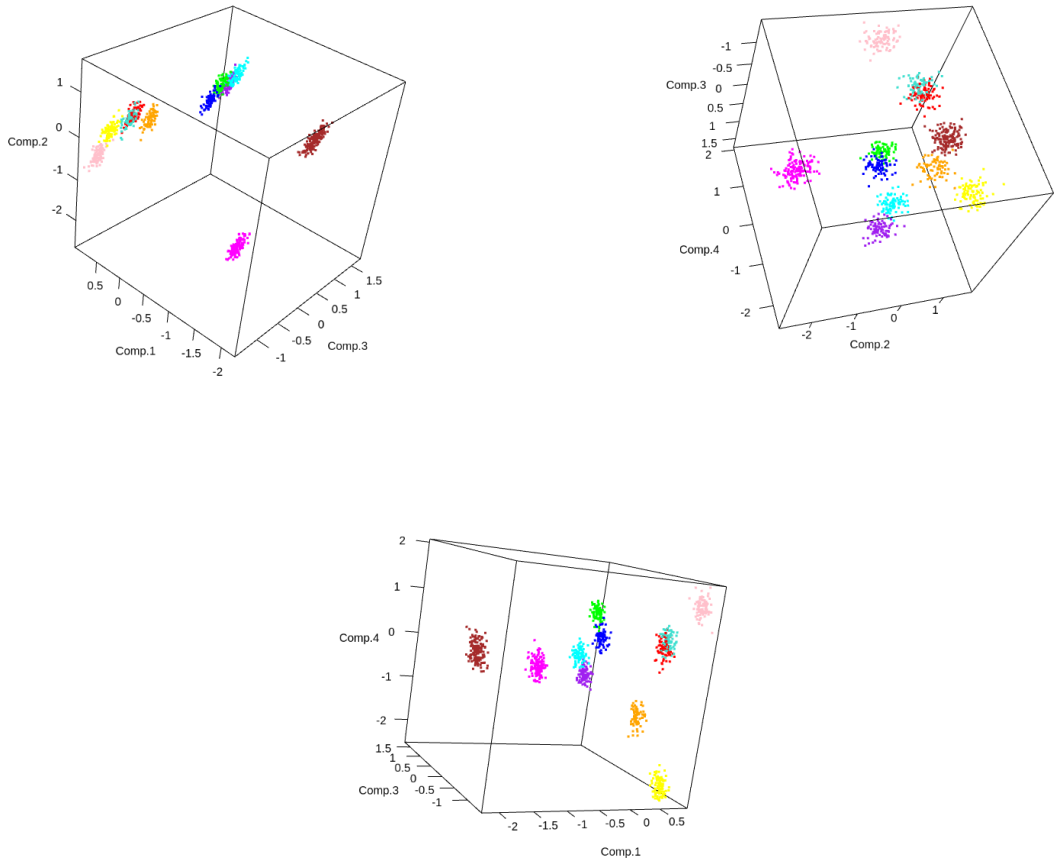

Рис. 13: Просторова діаграма розсіювання на перші чотири канонічні компоненти.

Тут складніше. Такого чіткого розділення, коли дивилися на проекції на головні компоненти, відстутнє. Здебільшого формуються невеликі кластери з двох-трьох підкластерів.

Кластеризація при $k = 14$.

```
> # Працюємо з кластеризацією з 14-и кластерів
> set.seed(777)
> data.kmeans <- kmeans(data.std, 14, nstart = 50)
> # Виводимо "звіт" по результатам процедури
> data.kmeans[ c("totss", "withinss", "tot.withinss", "betweenss")]

$totss
[1] 49950

$withinss
 [1] 114.18353 155.78017 128.18644 108.25493 102.74224  58.91698 116.87994
 [8] 130.96806 140.89864 118.68386  69.36651 125.41246 118.47443 114.09697

$tot.withinss
[1] 1602.845

$betweenss
[1] 48347.15
```

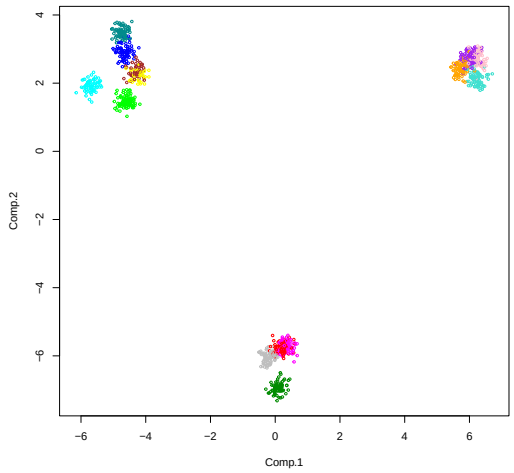

Рис. 14: Діаграма розсіювання даних на перші дві головні компоненти.

Навряд чи розмітка на двовимірній картинці допоможе нам, хоча більш-менш видно що відокремилося.

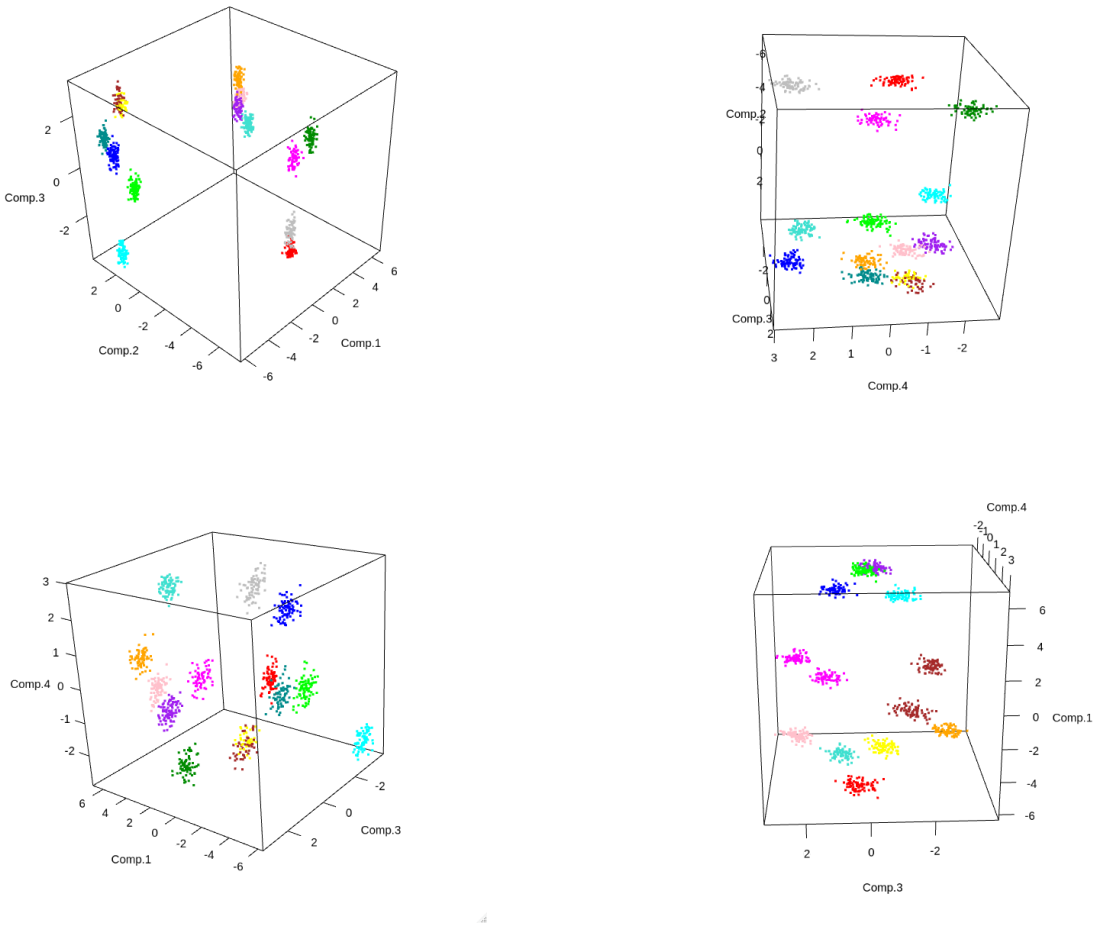

Рис. 15: Просторова діаграма розсіювання на перші чотири головні компоненти.

Тут можна вже побачити ”змішування” кластерів для цього випадку.

Побудуємо діаграму розсіювання даних зі зменшенням розмірності на основі методу канонічних компонент.

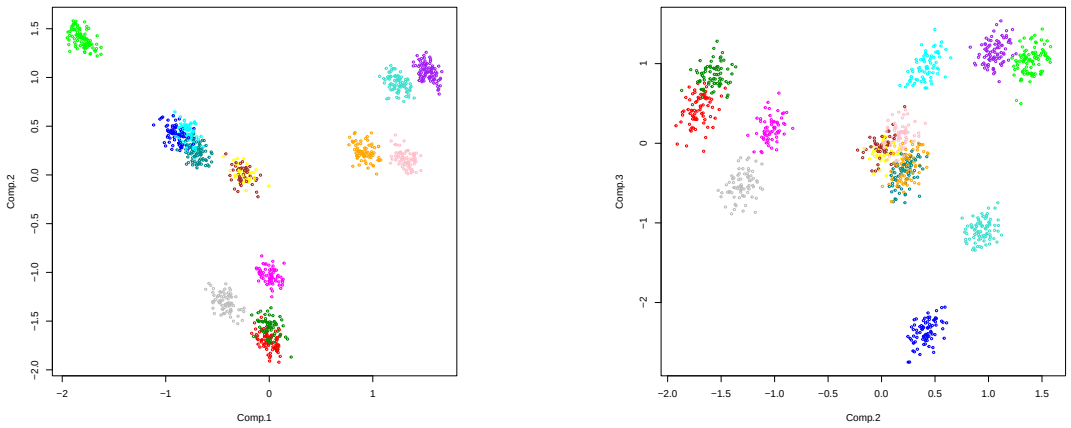

Рис. 16: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

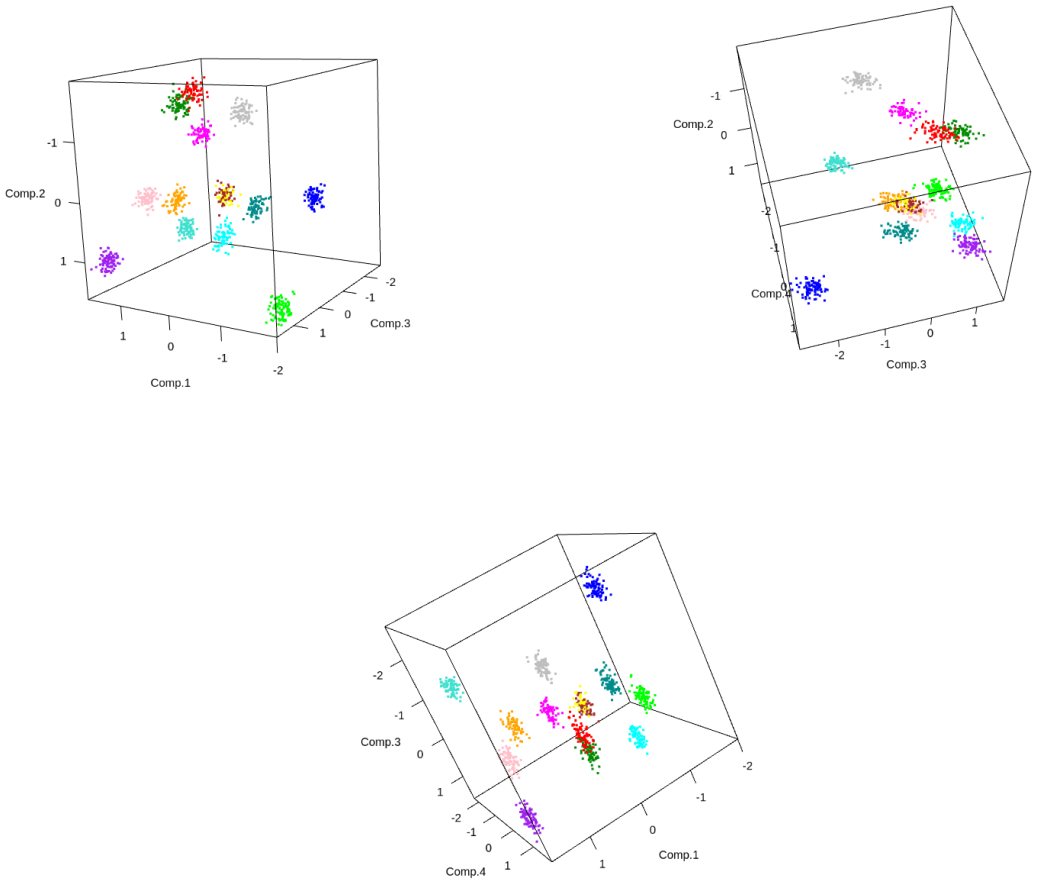

Рис. 17: Просторова діаграма розсіювання на перші чотири канонічні компоненти.

З кожним разом все починає зливатися у невідому кашу на проекціях. Тому дещо сумнівно використовувати такі кластеризації. А з іншого боку, зі збільшенням ”розмірності” розбиття, перегляду суто просторових діаграм починає давати менш інформативні результати. Знову ж таки, дивитися в високих розмірностях не виходить, тому поки змушені довіритися числовим показникам.

Метод медоїдів.

Тепер подивимося на те, що може дати кластеризація на основі методу медоїдів. Здається, що результат має вийтим приблизно таким самим, як і за методом центроїдів, бо дані стандартизовано. Надалі використовуємо реалізацію методу на основі функції *pam* з пакету *cluster*. Обчислення всі починаються з вибору оптимальної кількості розбиттів: у цьому випадку обмежимося висновками на основі середніх силуетів:

```
> fviz_nbclust(
+   x = data.std,
+   FUNcluster = pam,
+   method = "silhouette",
+   k.max = K.max
+ )
```

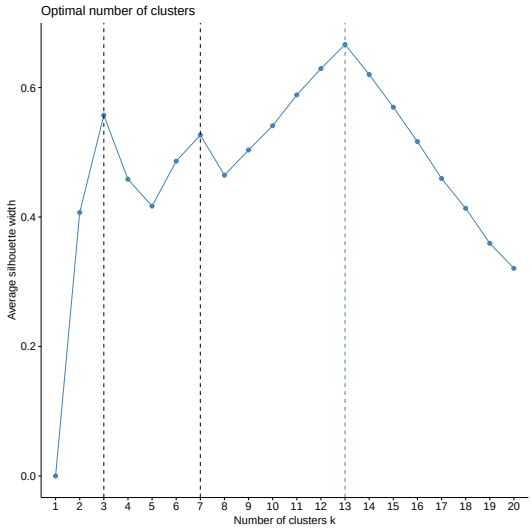

Рис. 18: Діаграма середніх силуетів. Кластеризація методом медоїдів.

З рисунка добре видно, що локальні максимуми утворюються при $k \in \{3, 7, 13\}$. Це трохи відрізняється від тих результатів, що виходили для методу центроїдів. Варто зауважити, що вигляд діаграми силуетів вирівнявся: у порівнянні з попереднім методом, тут немає ефекту шершавості (наявність коливань зі збільшенням кроку). Оберемо найкращу кластеризацію на основі візуалізацій аналогічними до попередніх досліджень.

Кластеризація при $k = 3$.

```
> # Працюємо з кластеризацією з 3-х кластерів
> set.seed(777)
> data.pam <- pam(data.std, 3, nstart = 50)
```

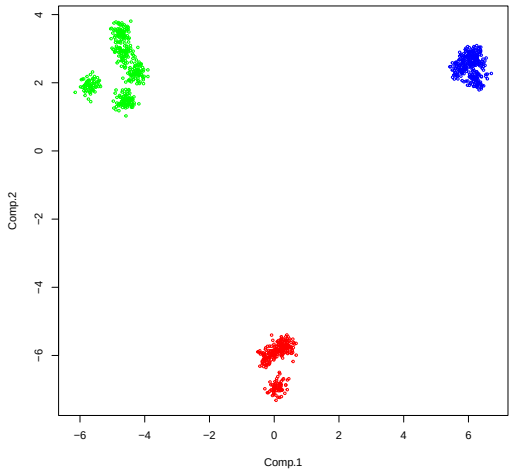

Рис. 19: Діаграма розсіювання даних на перші дві головні компоненти.

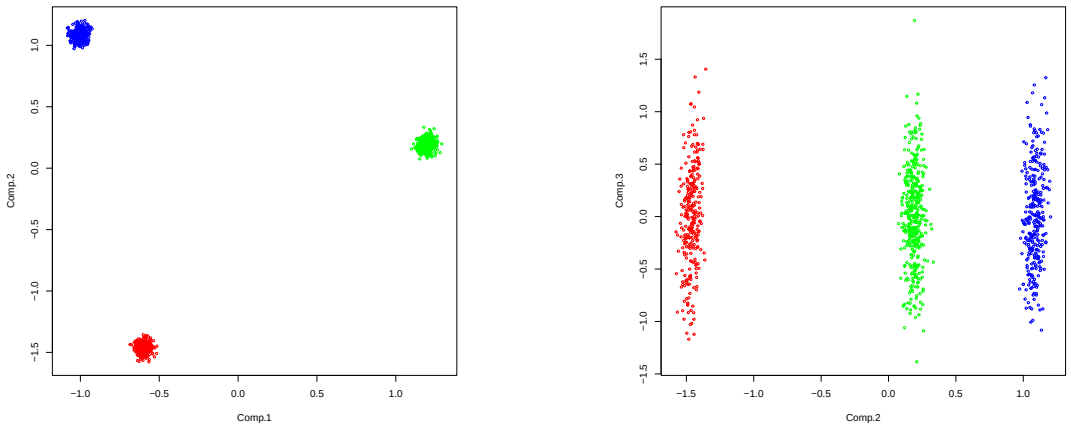

Рис. 20: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

Результати аналогічна до того, що виходило у методі центроїдів. Рухаємося далі.

Кластеризація при $k = 7$.

```
> # Працюємо з кластеризацією з 7-х кластерів
> set.seed(777)
> data.pam <- pam(data.std, 7, nstart = 50)
```

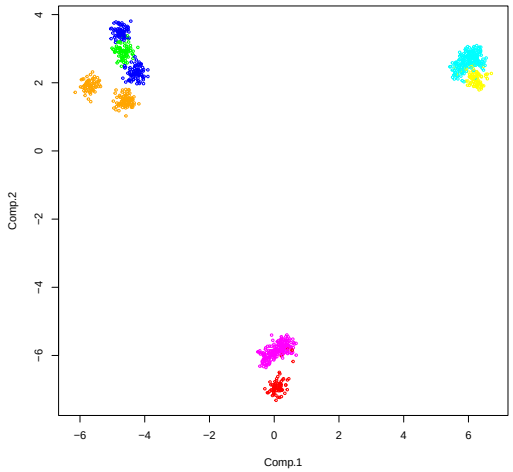

Рис. 21: Діаграма розсіювання даних на перші дві головні компоненти.

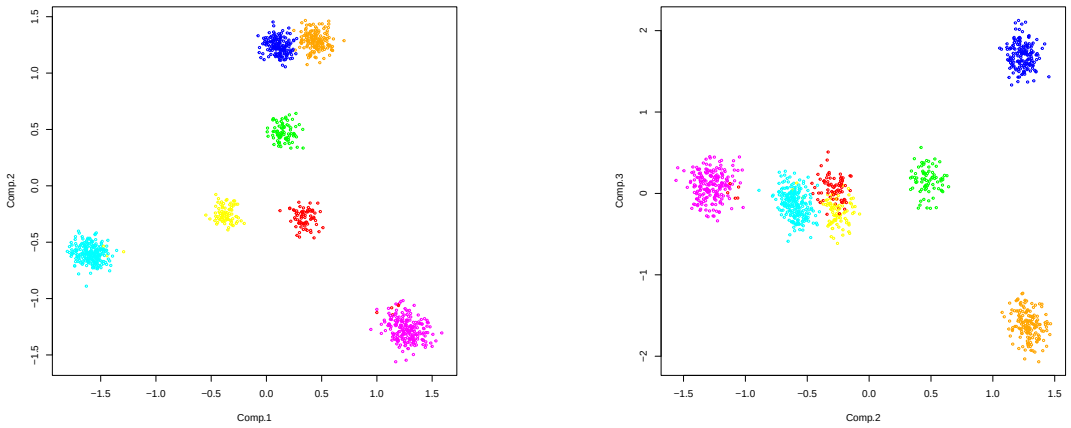

Рис. 22: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

На виході маємо дуже схожу ситуацію як у випадку методу центроїдів з шістьма кластерами. Подивимося на просторові діаграми.

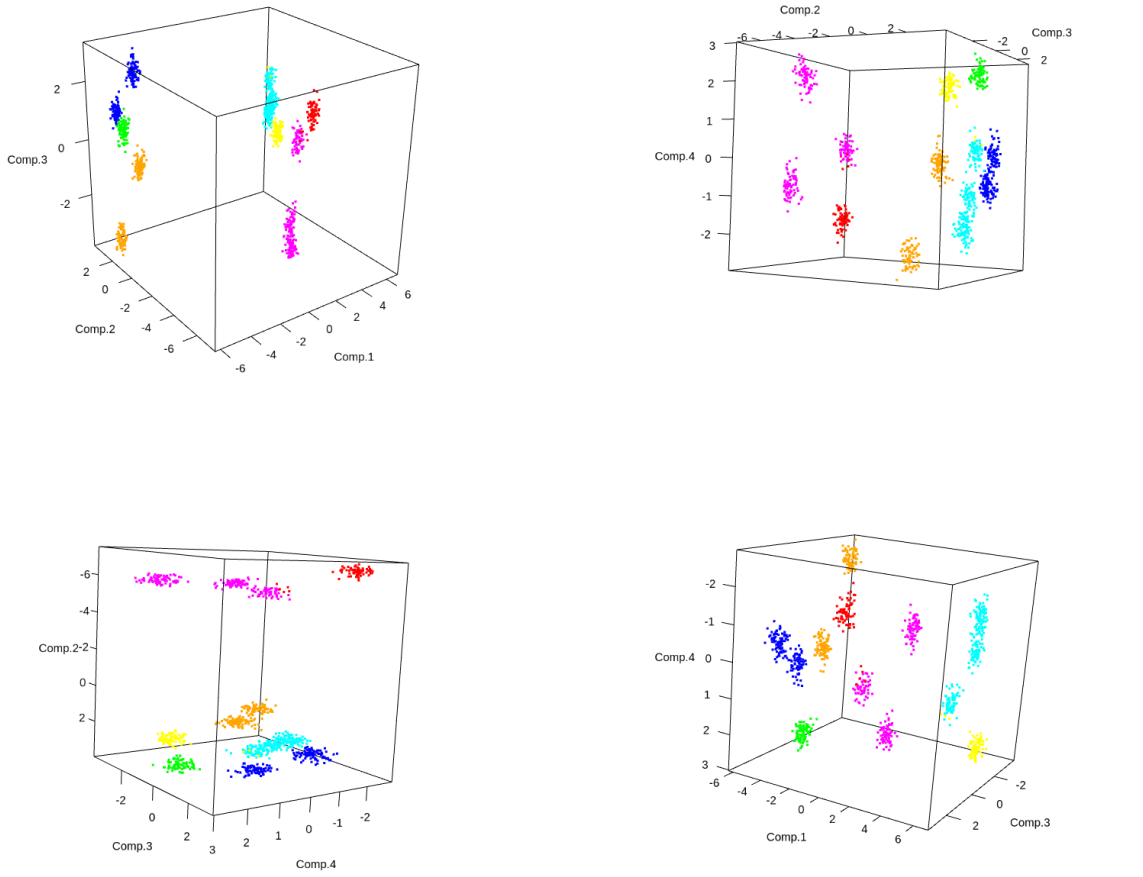

Рис. 23: Просторова діаграма розсіювання на перші чотири головні компоненти.

Виглядає кульгаво. Там, де чітко видно розділення між кластерами, можна помітити що деякі точки з одного кластера раптом відмітили до іншого. Неприємного ефекту змішування немає, але ця кластеризація викликає сумніви.

Кластеризація при $k = 13$.

```
> # Працюємо з кластеризацією з 13-и кластерів
> set.seed(777)
> data.pam <- pam(data.std, 13, nstart = 50)
```

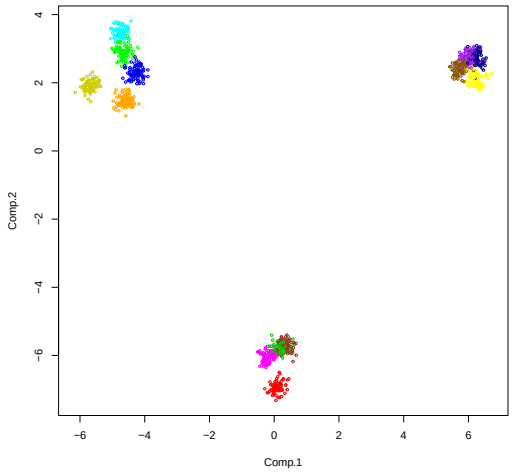

Рис. 24: Діаграма розсіювання даних на перші дві головні компоненти.

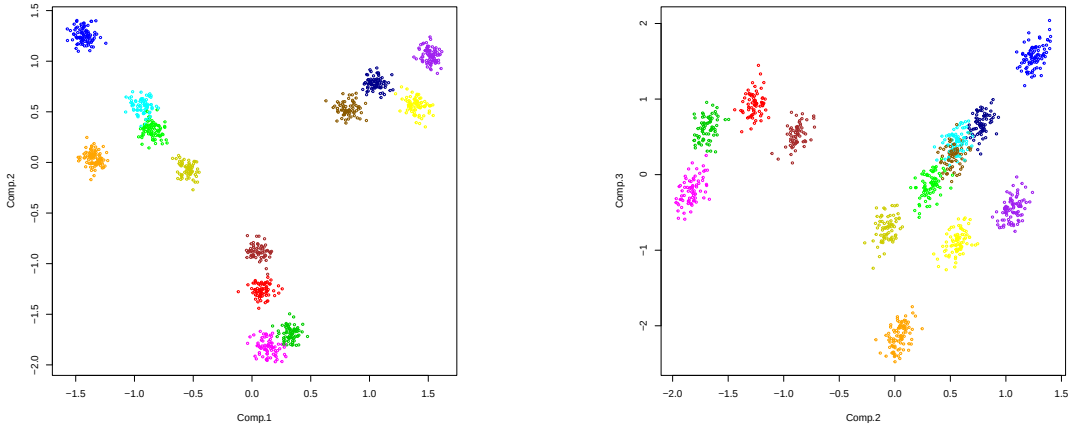

Рис. 25: Діаграма розсіювання даних на перші три канонічні компоненти. На основі пар канонічних компонент (1,2) та (2,3).

Тут картина більш приємна на око, бо змішування не таке серйозне, як спостерігалося на більших кластеризаціях раніше. Але варто підглянути на ситуацію з тривимірними діаграми розсіювання.

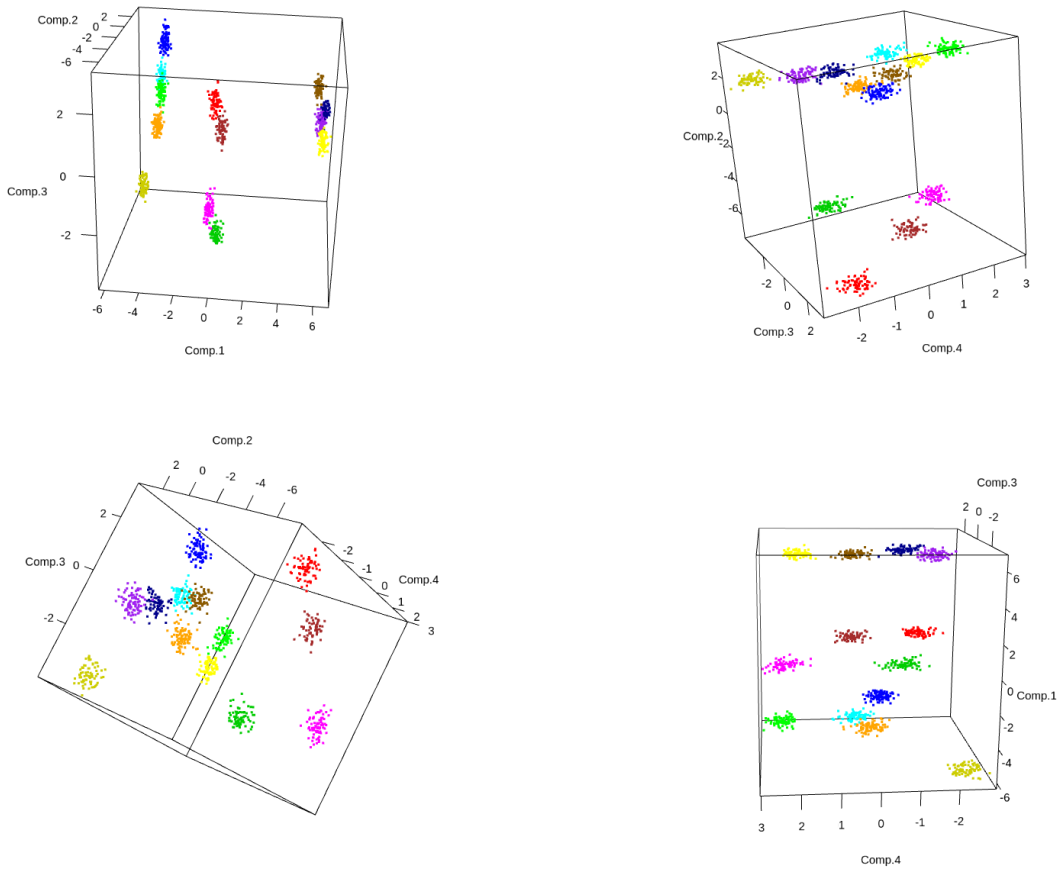

Рис. 26: Просторова діаграма розсіювання на перші чотири головні компоненти.

Виглядає добре. Спостерігаємо одне змішування для кластерів ”темно-синій” та ”фіолетовий”. Кластеризація виглядає узгодженою на основі проектувань на головні компоненти. А яка ситуація з проекціями на канонічні компоненти?

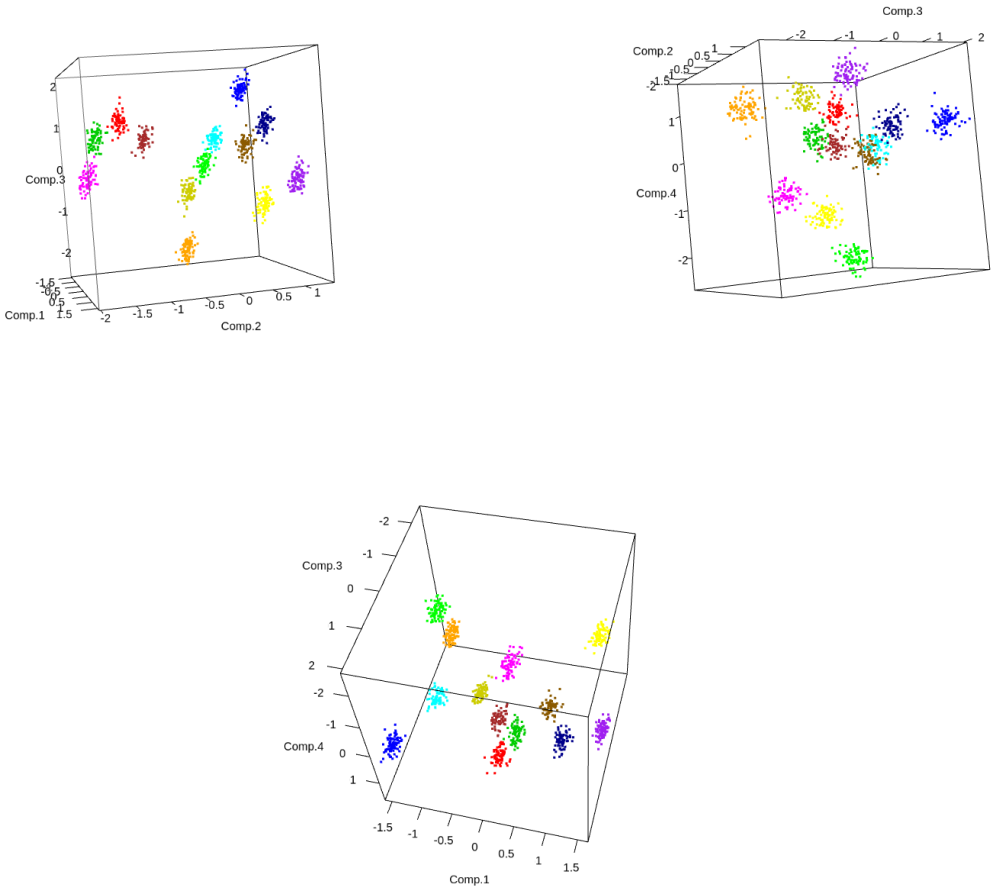

Рис. 27: Просторова діаграма розсіювання на перші чотири канонічні компоненти.

Трохи перемішалися інші декілька кластерів (але ситуація не така як на проекціях на головні компоненти), однак в цілому картинка нормальна. Порівнюючи інші кластеризації більшої розмірності, то в даному разі вийшло (відносно) найкращим чином.

Індекс Ренда на "хороших" розбиттях.

```
> # Працюємо з кластеризацією з 3-х кластерів
> set.seed(777)
> data.kmeans <- kmeans(data.std, 3, nstart = 50)
> data.pam <- pam(data.std, 3, nstart = 50)
> # Виводимо індекс Ренда для отриманих кластеризацій
> rand.index(data.pam$clustering, data.kmeans$cluster)
[1] 1
> table(data.pam$clustering, data.kmeans$cluster)
     1   2   3
  1 284   0   0
  2   0   0 404
  3   0 312   0
> # Працюємо з кластеризацією з 13-и кластерів
> set.seed(777)
> data.kmeans <- kmeans(data.std, 13, nstart = 50)
> data.pam <- pam(data.std, 13, nstart = 50)
> # Виводимо індекс Ренда для отриманих кластеризацій
> rand.index(data.pam$clustering, data.kmeans$cluster)
[1] 0.9841722
> table(data.pam$clustering, data.kmeans$cluster)
     1  2  3  4  5  6  7  8  9 10 11 12 13
  1  0  0  0  0  0  0  0  0  0  0  0  0 73
  2  0  0  0  0  0  0  0  0  0  0 79  0  0
  3  0  0  0  0  0  0  0  0 88  0  0  0  0
  4  0  0  0  0  0  0 91  0  0  0  0  0  0
  5  0  0  0  0  0  0  0  0  0  0  0 74  0
  6  0  0  0  0 82  0  0  0  0  0  0  0  0
  7  0  0  0  0  0  0  0  0 74  0  0  0  0
  8  0  0  0 85  0  0  0  0  0  0  0  0  0
  9  0  0  0  0  0  0  0 73  0  0  0  0  0
  10 41  0  0  0  0 34  0  0  0  0  0  0  0
  11  0 64  0  0  0  0  0  0  0  0  0  0  0
  12  0  0  0  0  0  0  0  0  0 72  0  0  0
  13  0  0 70  0  0  0  0  0  0  0  0  0  0
```

Висновки.

На діаграмах можна було помітити чітке розмежування даних на три великі кластери. Для відображення загальних тенденцій, дорчено використовувати кластеризацію з потрійним розбиттям відповідно. Вибір алгоритму формування кластерів для даної ситуації неважливий – результат майже однаковий.

Для розмітки даних, скажімо, на детальні складові, варто застосувати кластеризацію на 13 груп. Як показали результати, метод медоїдів непогано впорався з задачею. Була думка, що метод центроїдів в цьому випадку може теж спрацювати нормально, оскільки дані зведені до стандартної шкали.

Частина друга.

Вступ.

Проведено кластеризацію за даними про солодкі напої в магазині. Використані методи відносно примітивні, бо розмірність даних невисока. Результатам кластеризації надано інтерпретацію.

Хід роботи.

Деякі відомості про дані.

Таблиця містить дані про 24 напої, які вдалося знайти автору під час одного з походів у магазин. Напої представляють з себе різновиди газованих напоїв, холодних чаїв та енергетиків. Для кожного напою зібрано відомості про споживчу цінність на 100 мл (а саме: енергетична цінність, білки, вуглеводи, жири) та ціна напою за 0.33 літра. Наведемо нижче цю таблицю (показувати колонки про наявність жирів та білків немає сенсу, бо там все по нулям).

```
> drinks.data <- read.csv("./drinks_100ml.csv", header=T)
> drinks.data[,c("Напій", "Енергетична.цінність", "Вуглеводи", "Сіль", "Ціна")]
```

	Напій	Енергетична.цінність	Вуглеводи	Сіль	Ціна
1	Arizona Tea: білий	87 кДж / 21 ккал	4.9	0.00	35.99
2	Arizona Tea: червоний	87 кДж / 21 ккал	5.0	0.00	35.99
3	Canada Dry	116 кДж / 27 ккал	6.5	0.00	27.99
4	Coca-Cola	180 кДж / 42 ккал	10.6	0.00	15.14
5	Dr Pepper	119 кДж / 28 ккал	6.9	0.00	29.28
6	Dr Pepper Cherry	120 кДж / 28 ккал	6.8	0.01	29.28
7	Fanta Orange	140 кДж / 33 ккал	7.9	0.00	14.74
8	Fanta Schokata	120 кДж / 28 ккал	6.8	0.04	14.74
9	Fanta мандарин	213 кДж / 50 ккал	12.3	0.00	14.74
10	Maaza Mango	208 кДж / 49 ккал	12.0	0.00	31.99
11	Mirinda Orange	179 кДж / 43 ккал	11.2	0.04	13.52
12	Non Stop Evo	140 кДж / 33 ккал	8.3	0.00	27.26
13	Non Stop Original	178 кДж / 42 ккал	10.0	0.00	17.06
14	Pepsi	182 кДж / 43 ккал	11.0	0.01	13.52
15	Royal Club: Ginger Ale	162 кДж / 39 ккал	9.4	0.00	25.24
16	Schweppes Grapefruit	99 кДж / 23 ккал	5.4	0.00	15.44
17	Schweppes Mochito	187 кДж / 44 ккал	10.6	0.02	15.44
18	Schweppes Pinacolada	139 кДж / 33 ккал	7.9	0.03	15.44
19	Schweppes Tonic	157 кДж / 37 ккал	8.9	0.00	15.44
20	Sprite	122 кДж / 29 ккал	7.0	0.02	14.74
21	Грузинський букет: тархун	192 кДж / 46 ккал	11.5	0.00	22.66
22	Живчик: яблуко + ехінацея	155 кДж / 37 ккал	9.6	0.01	13.20
23	Караван: ситро	157 кДж / 37 ккал	9.6	0.00	5.33
24	Розмай лісовий ситро	182 кДж / 43 ккал	10.7	0.00	12.99

Зосередимося на двовимірному представленні даних, а саме на кількості вуглеводів та ціні. Зобразимо дані на діаграмі розсіювання.

```
> drinks.data.reduced <- drinks.data[,c("Вуглеводи", "Ціна")]
```

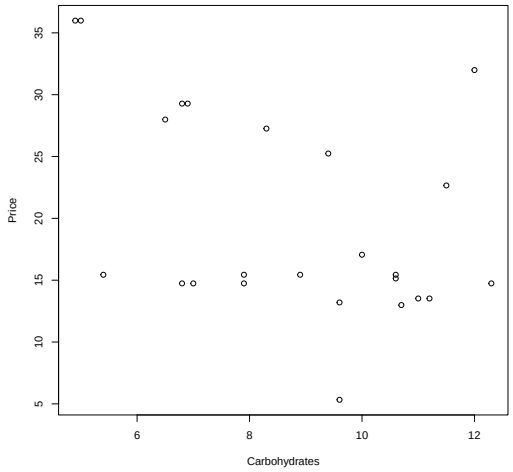

Рис. 28: Діаграма розсіювання за даними про напої.

На діаграмі можна побачити певне розмежування за ціною: виділяється група напоїв з високою ціною за 0.33 літра та з меншою відповідно. Застосуємо кластеризацію за методом центроїдів.

```
> fviz_nbclust(
+   x = drinks.data.reduced,
+   FUNcluster = kmeans,
+   method = "wss",
+   k.max = 10
+ )
> fviz_nbclust(
+   x = drinks.data.reduced,
+   FUNcluster = kmeans,
+   method = "silhouette",
+   k.max = 10
+ )
```

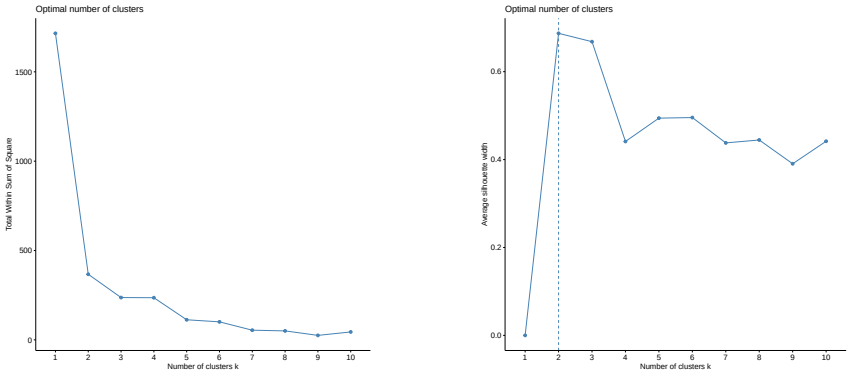

Рис. 29: Графік сумарного внутрішньокластерних відхилень. Діаграма середніх силуетів.

Найбільший силует виходить для кластеризацій на 2 чи 3 частини. Злам маємо для кластеризації на 2 частини. Для інших розбиттів середній силует виходить замалим, що робить сумнівним можливість використання більшої кількості кластерів. Розглянемо що виходить для двох кластерів:

```
> # Працюємо з кластеризацією з 2-х кластерів
> set.seed(777)
> drinks.kmeans.2 <- kmeans(drinks.data.reduced, 2, nstart = 25)
> # Виводимо "звіт" по результатам процедури
> drinks.kmeans.2[ c("totss", "withinss", "tot.withinss", "betweenss")]

$totss
[1] 1715.553

$withinss
[1] 150.6318 216.5200

$tot.withinss
[1] 367.1517

$betweenss
[1] 1348.401
```

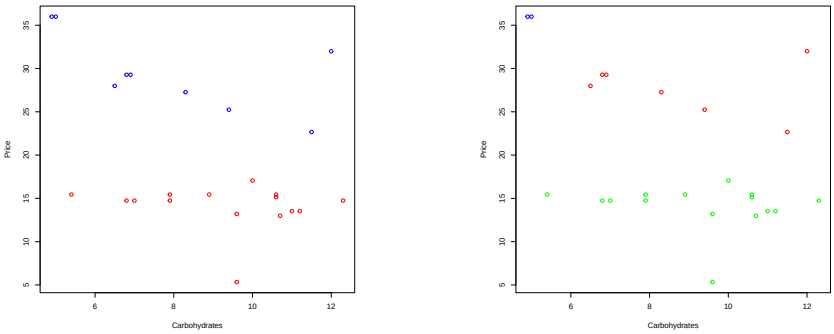

Рис. 30: Діаграма розсіювання за даними про напої. Кластери виділено кольорами.

Припущення про розмежовування за ціною підтверджується, кластери містять напої з відповідним рівнем ціни.

Висновки.

Побудовані кластери можна інтерпретувати з логічних міркувань. Єдине що можна було б зробити – зібрати додаткові дані про напої (наприклад, колір напою, прозорість тощо), аби можна було б помітити можливі особливості тих чи інших груп.

Зауваження. Автор не вживав напої, що наведені у таблиці. Треба ж ще й на здорове тіло писати звіт.