ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4
«ГОЛОВНІ КОМПОНЕНТИ І СПЕКТРАЛЬНА КЛАСТЕРИЗАЦІЯ»

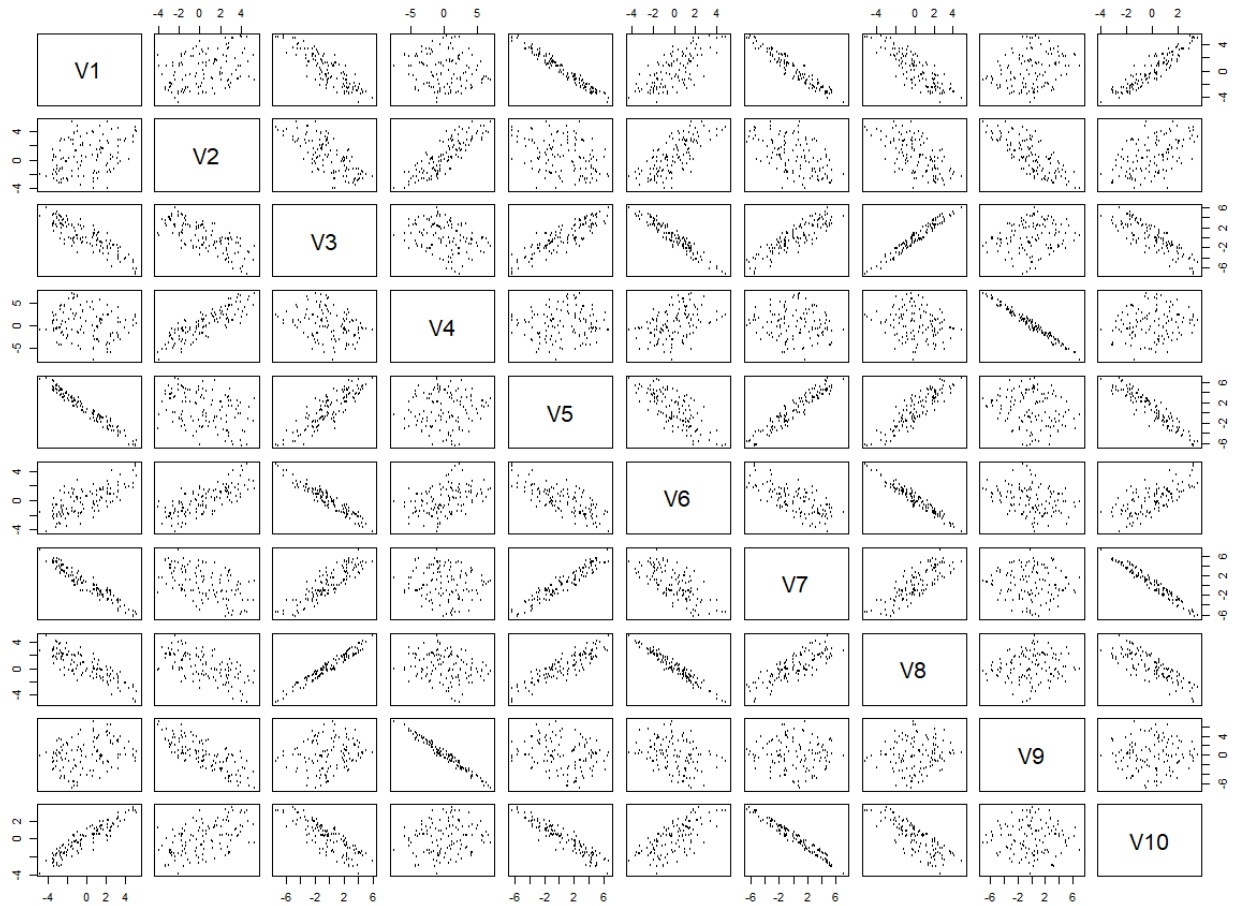
частина 1

Ломако О., 2 к. маг, «статистика», варіант 9

Нехай маємо певні змодельовані дані, що містяться у файлі F9.txt. Для них спробуємо розв'язати задачу кластеризації, використовуючи техніку спектральної кластеризації.

Для цього, спершу, звичайно, зчитаємо дані і виведемо попарні діаграми розсіювання.

```
> # зчитуємо дані
> data <- read.table('C:\\Users\\Razor\\Desktop\\дистанційне навчання\\статис
тичний аналіз багатовимірних даних\\lab4\\F9.txt')
> # зобразимо діаграму розсіювання пар
> pairs(data, cex = 0.1)
```

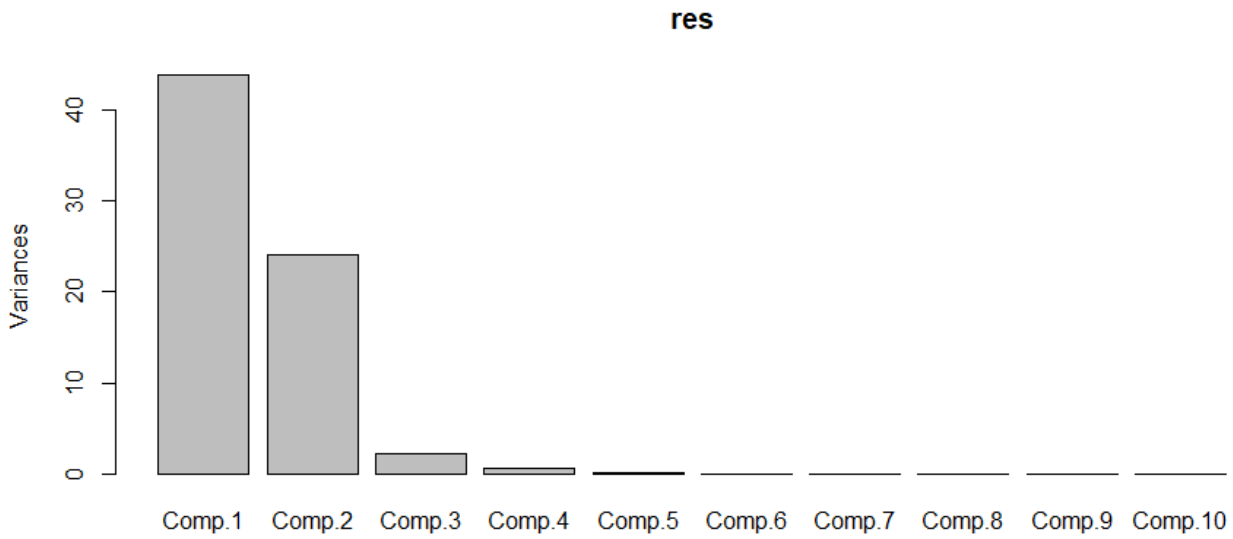


Попарні діаграми розсіювання не дають чіткої відповіді щодо якоїсь чіткої геометричної структури даних: десь на діаграмі точки разом утворюють «спадну» пряму, десь – «зростаючу».

Тому спробуємо знайти приховану геометричну структуру, використовуючи метод головних компонент.

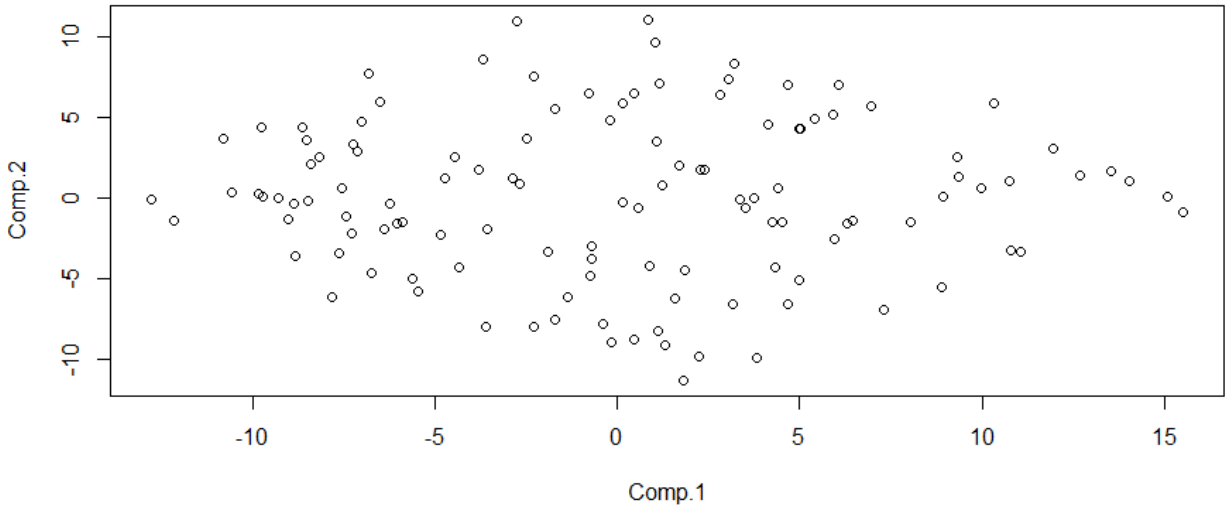
```
> # застосуємо метод головних компонент
> res <- princomp(data)
```

```
> plot(res)
> summary(res)
Importance of components:
                          Comp.1    Comp.2     Comp.3      Comp.4       Comp.5       Comp.6       Comp.7 Comp.8
Standard deviation     6.6149146 4.9082314 1.51243113 0.811654809 0.280875609 5.038853e-08 4.620431e-09      0
Proportion of Variance 0.6174019 0.3399144 0.03227533 0.009295274 0.001113134 3.582472e-17 3.012205e-19      0
Cumulative Proportion  0.6174019 0.9573163 0.98959159 0.998886866 1.000000000 1.000000e+00 1.000000e+00      1
                       Comp.9 Comp.10
Standard deviation          0       0
Proportion of Variance      0       0
Cumulative Proportion       1       1
```



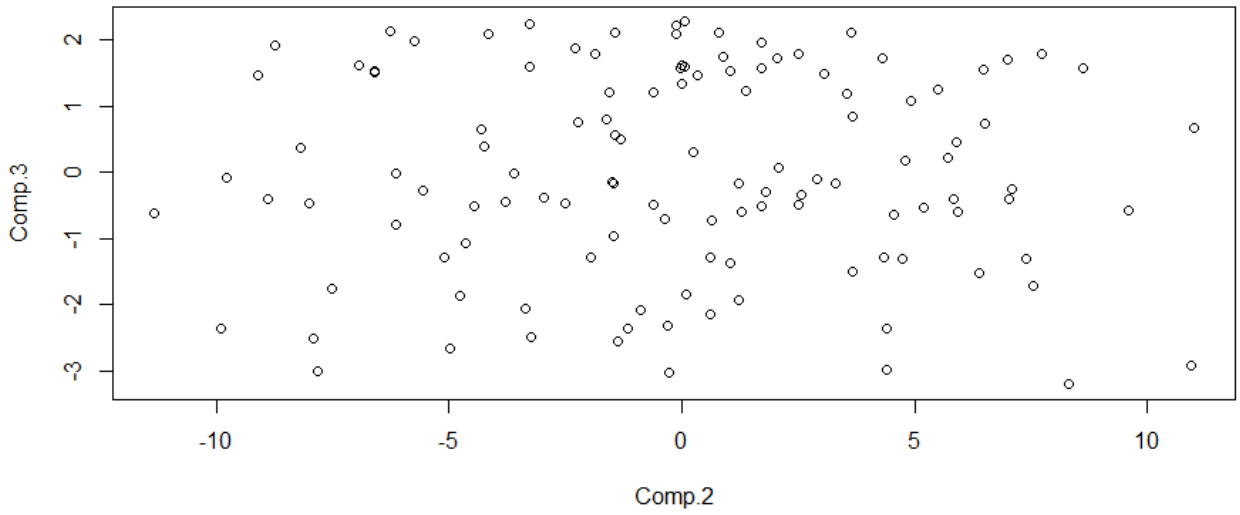
Так, перша компонента пояснюватиме 61.7% розкиду даних, в той час як перші дві – вже 95.7%. При цьому з діаграми власних чисел доцільно для аналізу взяти саме перші дві компоненти, адже саме після другої відбувається злам.

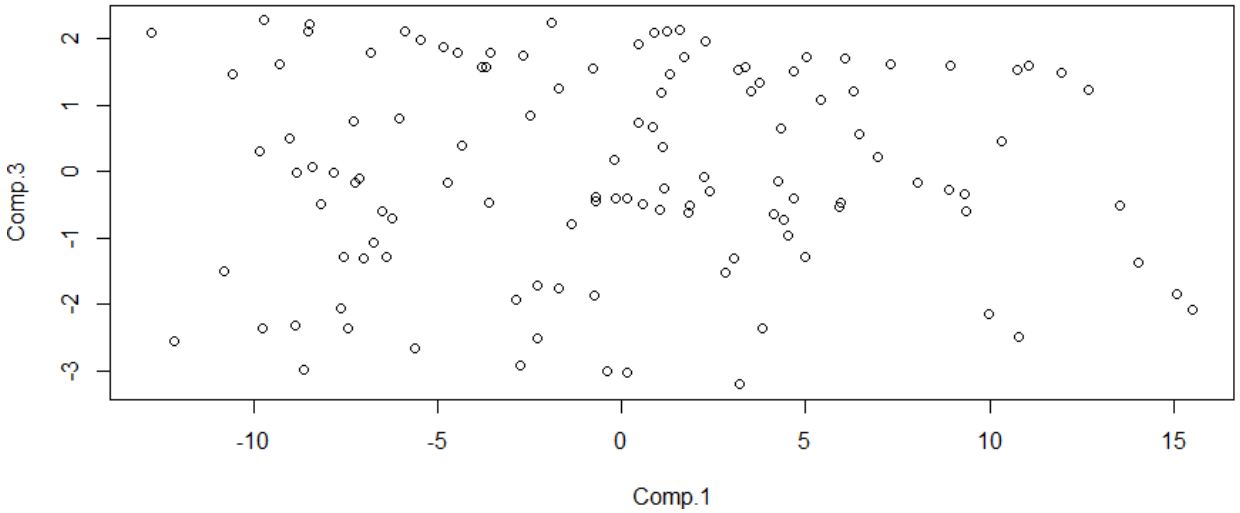
```
> # виведемо діаграму розсіювання перших двох головних компонент
> plot(res$scores[,1:2])
```



Така діаграма розсіювання ніякої геометричної структури не виділяє, тому до розгляду залучимо і третю головну компоненту. Поглянемо на всі попарні діаграми розсіювання для перших трьох головних компонент.

```
> # виведемо попарні діаграми розсіювання перших трьох головних компонент
> plot(res$scores[,2:3])
> plot(res$scores[,c(1,3)])
```

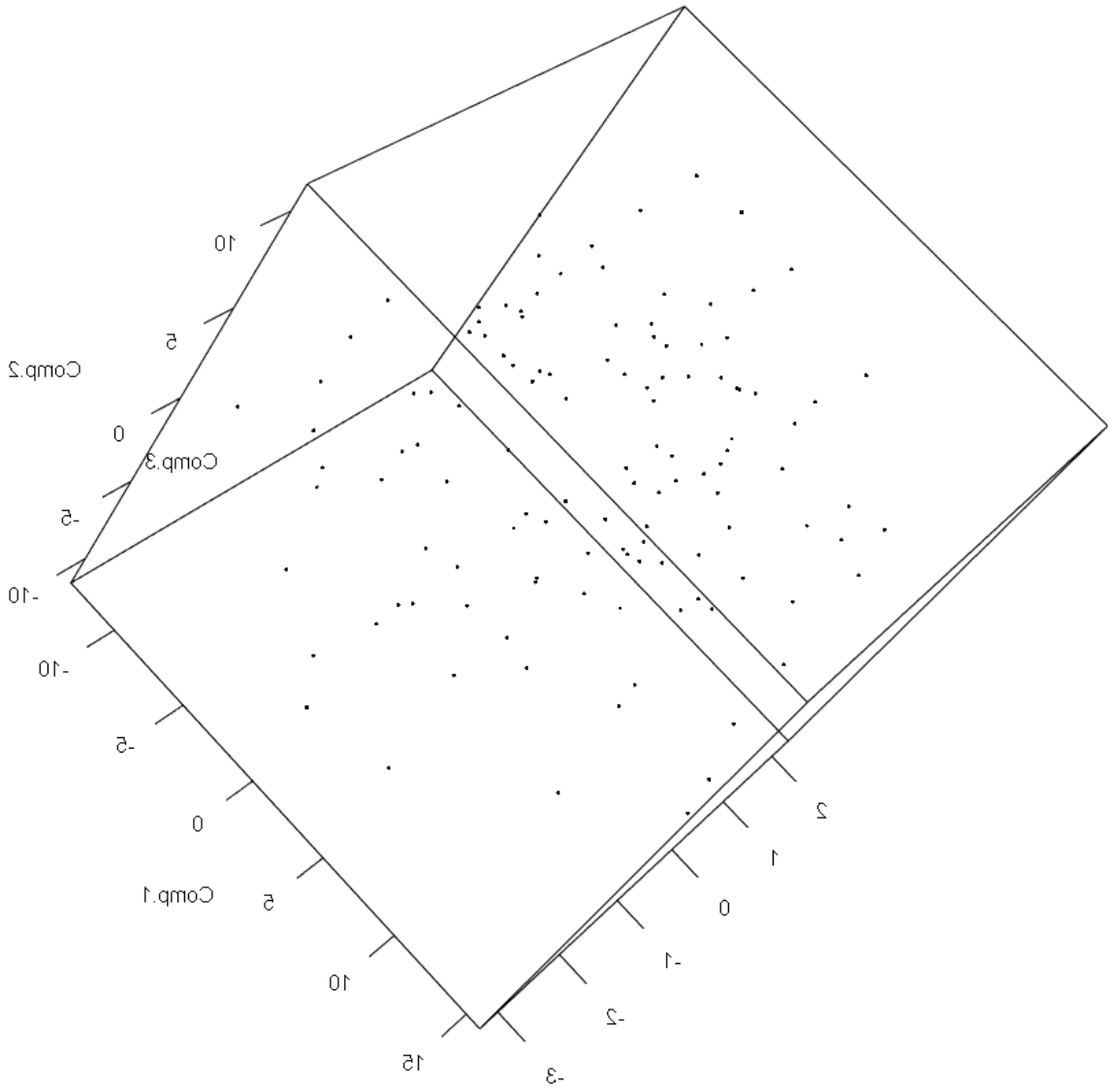




Разом з минулою діаграмою, ці дві, знову ж таки, не дають особисто мені ніякої конкретики в плані якої-небудь геометричної структури.

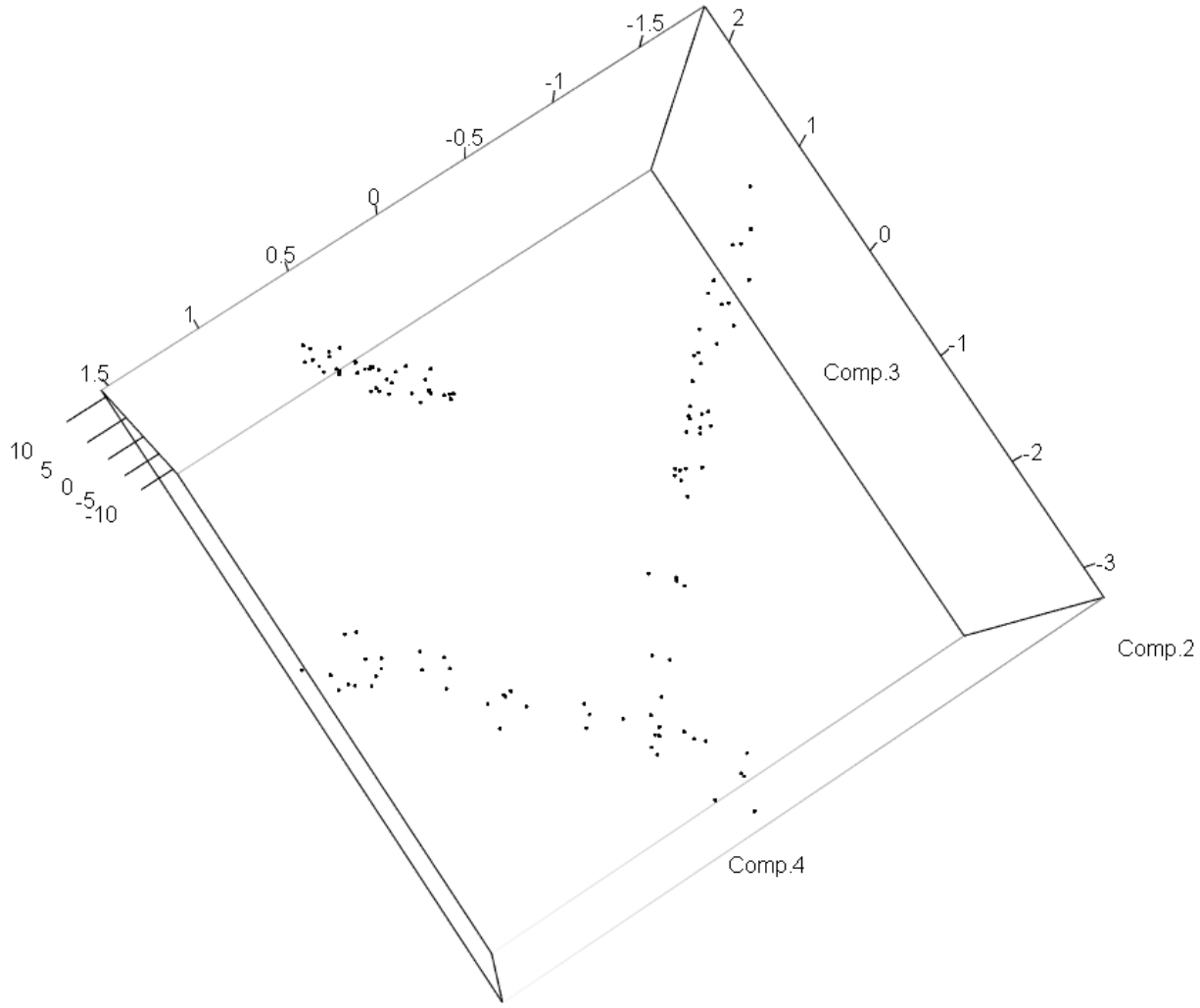
Спробуємо покрутити в тривимірному просторі тривимірну діаграму.

```
> # виведемо тривимірну діаграму розсіювання
> library(rgl)
> plot3d(res$scores[,1:3])
```



На жаль, навіть після численних поворотів тривимірного зображення, віднайти бодай яку-небудь геометричну структуру не вдалося. Тому спробуємо поглянути на тривимірну діаграму, наприклад, 2, 3 і 4-ї компонент.

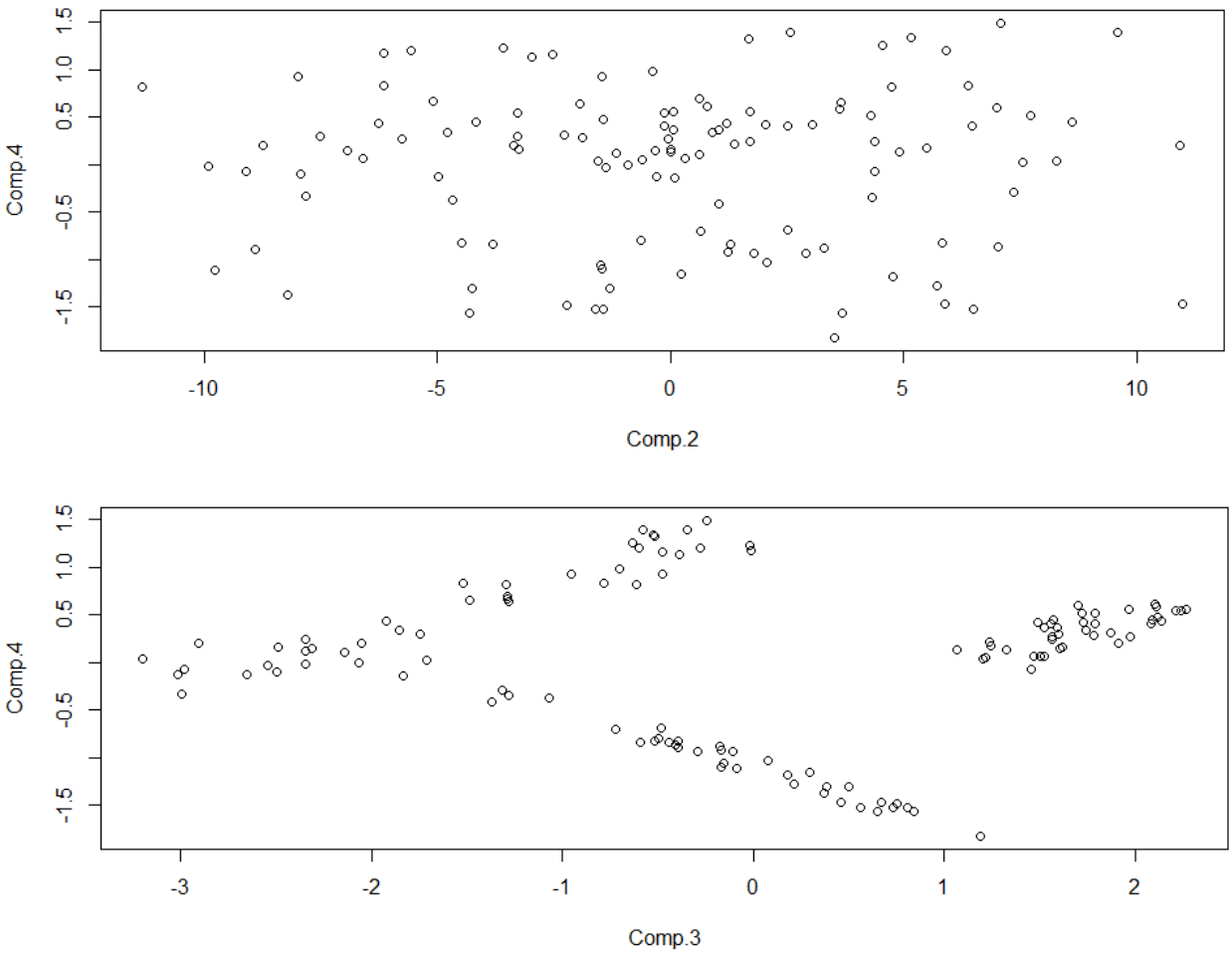
```
> plot3d(res$scores[,2:4])
```



Тут же можемо бачити, що певна структура має місце бути: явно виділяється одна купка зліва зверху, а також знизу і справа зверху виділяються дві купки точок. Єдине питання в невеликій групці точок між ними, але, здається, на цій тривимірній діаграмі можна виділити 3 кластери.

Поглянемо на попарні діаграми розсіювання другої та третьої компонент з четвертою.

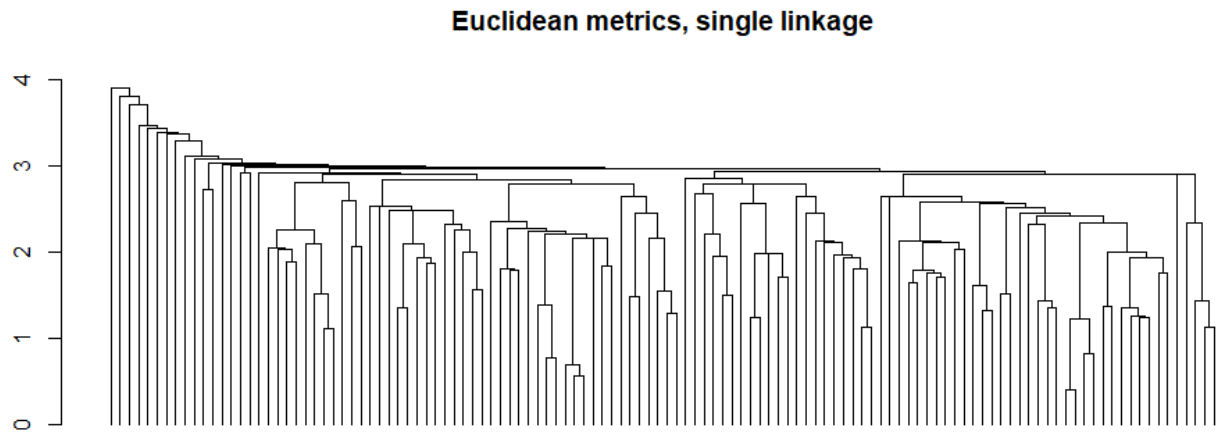
```
> plot(res$scores[,c(2,4)])
> plot(res$scores[,3:4])
```

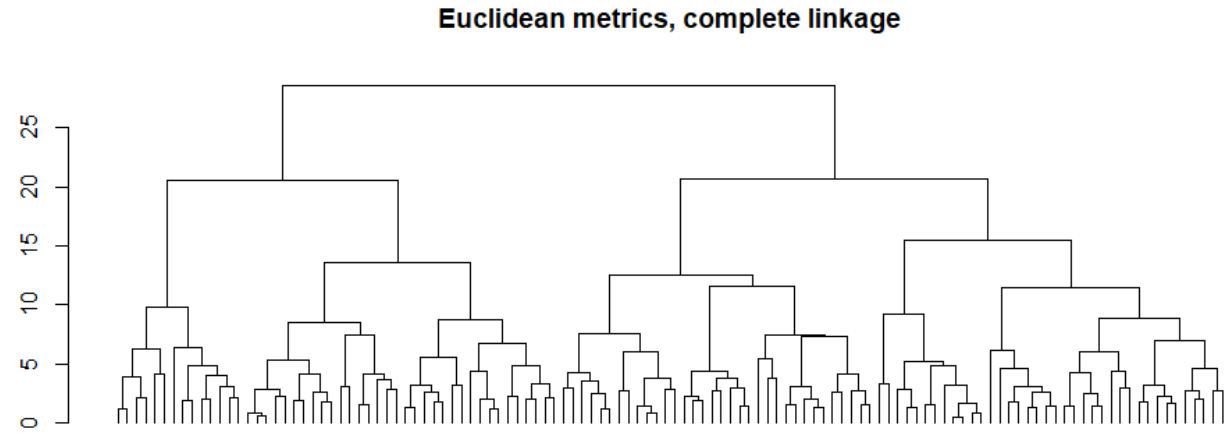


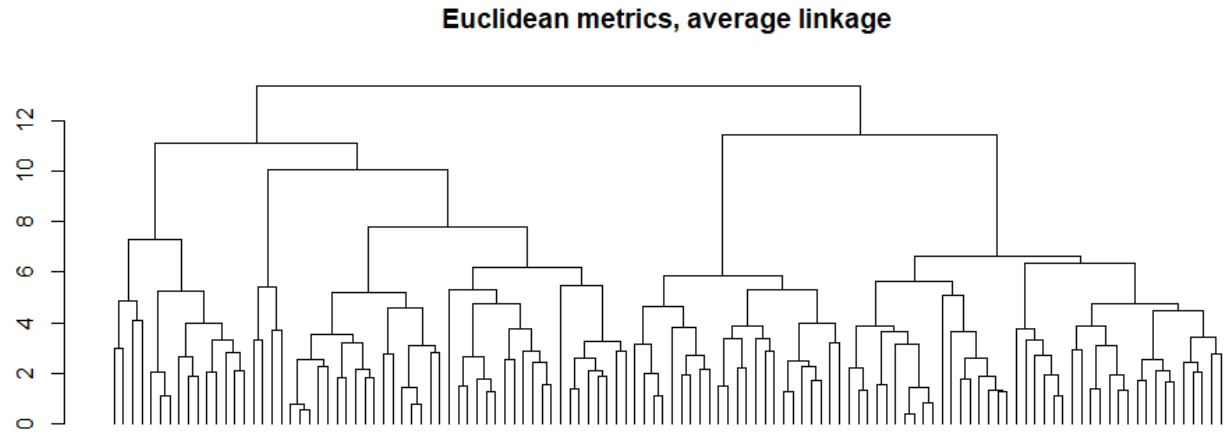
Остання двовимірна діаграма є найбільш красномовною – вона і підтверджує наші догадки про бодай яку-небудь геометричну структуру. Хоча, на ній виділити саме три кластери я зі 100-%-ою впевненістю не можу.

Далі для початкових даних проведемо ієрархічний кластерний аналіз. Спершу використовуватимемо евклідову відстань, і спробуємо застосовані в 3-й роботі методи (одного, повного та середнього зв’язку).

```
> plot(as.dendrogram(hclust(d, method = 'single')), leaflab = 'none', main =
'Euclidean metrics, single linkage')
> plot(as.dendrogram(hclust(d, method = 'complete')), leaflab = 'none', main
= 'Euclidean metrics, complete linkage')
> plot(as.dendrogram(hclust(d, method = 'average')), leaflab = 'none', main =
'Euclidean metrics, average linkage')
```



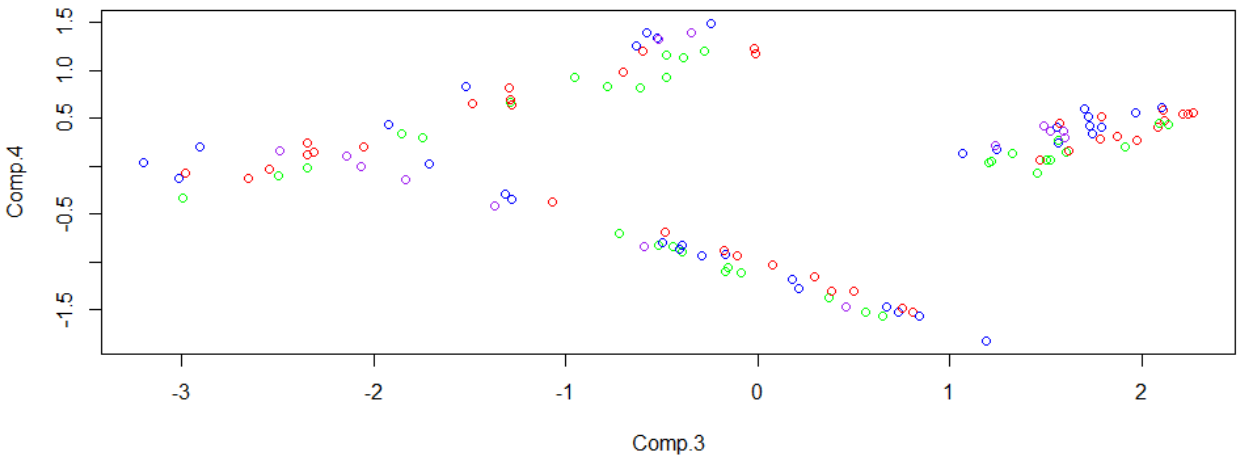




Метод одного зв’язку взагалі не дає ніякого уявлення про можливу кластеризацію. Метод же повного зв’язку достатньо чітко показує нам наявність чотирьох кластерів, в той час як метод середнього зв’язку, в певній мірі, збільшує їх кількість до п’яти.

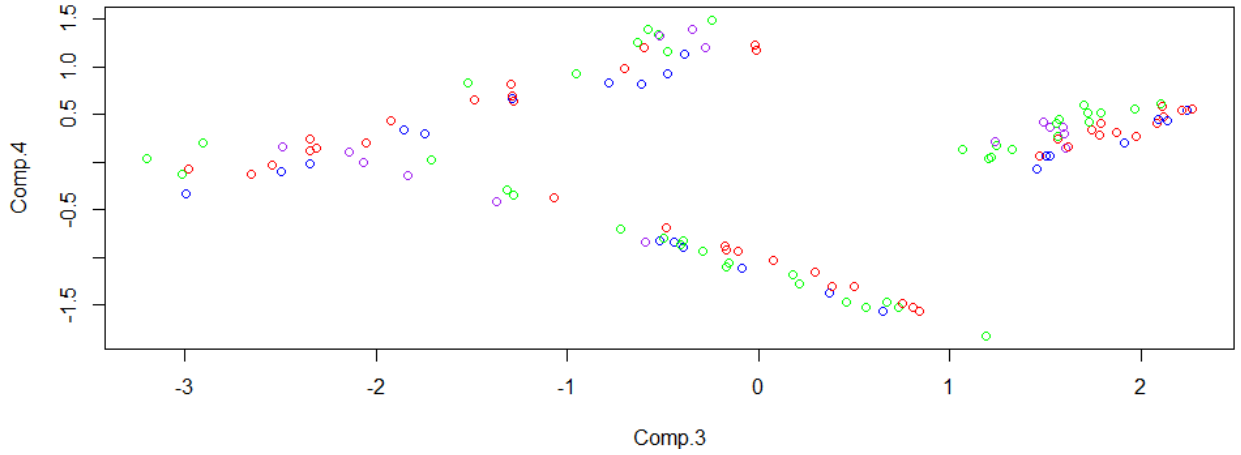
Спробуємо витягнути з методу повного зв'язку кластеризацію на 4 кластери, і подивитись, як відбудеться розфарбування 3 і 4 головних компонент (нагадаю, бодай яку-небудь структуру вдалось побачити лише там).

```
> groups1 <- cutree(hclust(d, method = 'complete'), k = 4)
> plot(res$scores[,c(3,4)], col = c('red', 'green', 'blue', 'purple')[groups1
])
```



Отримали абсолютно і повністю не те, на що розраховували. Поглянемо на варіант, отриманий методом середнього зв'язку.

```
> groups2 <- cutree(hclust(d, method = 'average'), k = 4)
> plot(res$scores[,c(3,4)], col = c('red', 'green', 'blue', 'purple')[groups2
])
```

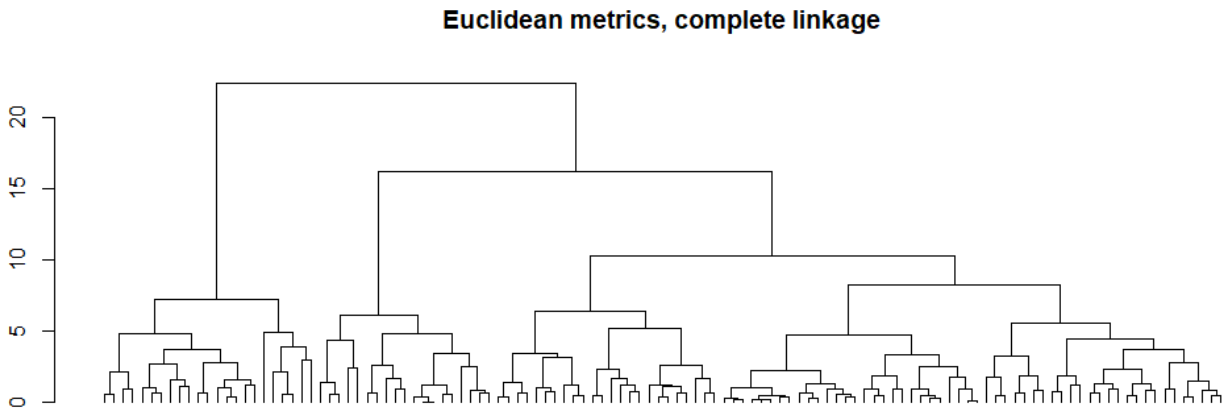


В принципі, очікуваний результат: така кластеризація є цілком невдалою.

Спробуємо тепер застосувати метод ієрархічної кластеризації до даних, які були отримані вже самим методом головних компонент. Але тут варто зауважити, що перші три головні компоненти пояснюватимуть розкид аж 98.95% даних, в той час як перші чотири пояснюватимуть аж 99.89%. Тому, здається, техніка ієрархічної кластеризації для тих же методів не принесе гарного результату. Зауважу, що аналізуватимемо 2-4 компоненти, оскільки саме в них вдалося виділити бодай яку-небудь геометричну структуру.

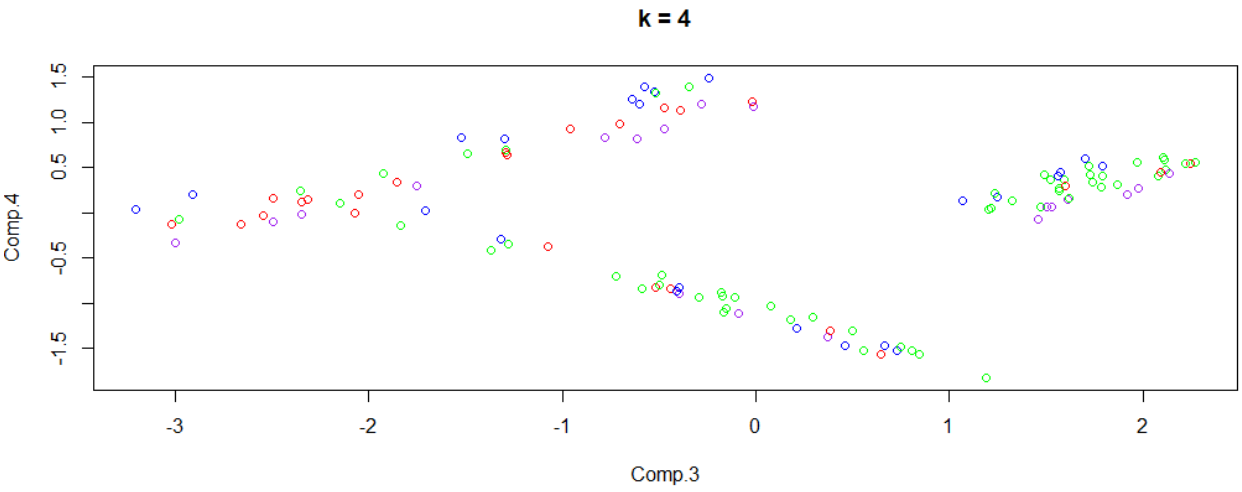
Спершу розглянемо евклідову метрику і метод повного зв'язку.

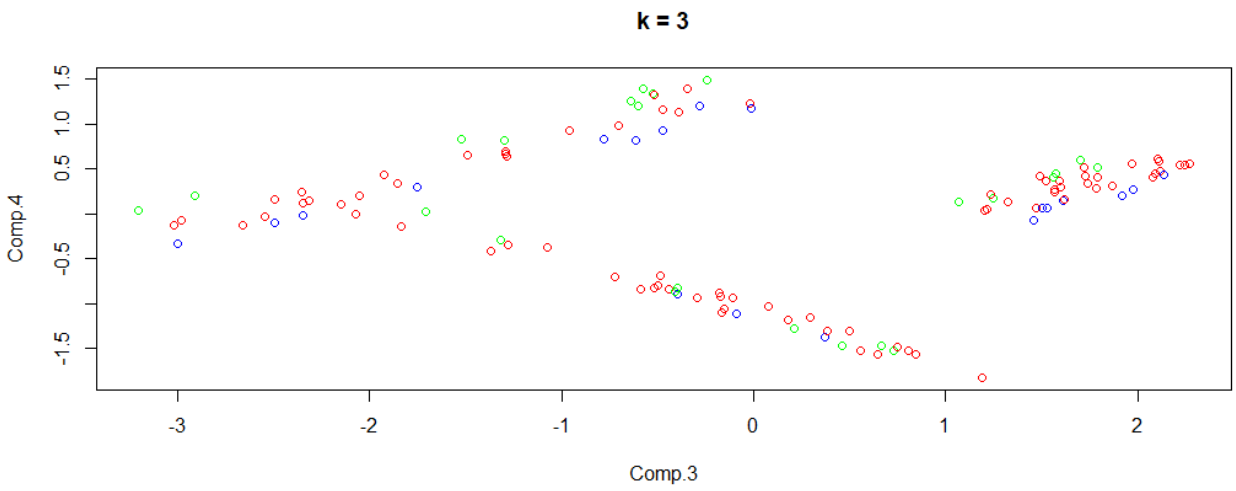
```
> pca_d_e <- dist(res$scores[,2:4], method = 'euclidean')
> plot(as.dendrogram(hclust(pca_d_e, method = 'complete')), leaflab = 'none',
+      main = 'Euclidean metrics, complete linkage')
```



Тут можна виділити від 2 до 5, здається, кластерів. Подивимось як розфарбується двовимірна діаграма розсіювання у просторі 3 і 4 головних компонент.

```
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green', 'blue', 'purple')[cutree(hclust(pca_d_e,
+                                                              method = 'comp
lete'), k = 4)],
+      main = 'k = 4')
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green', 'blue')[cutree(hclust(pca_d_e,
+                                                    method = 'comp
lete'), k = 3)],
+      main = 'k = 3')
```

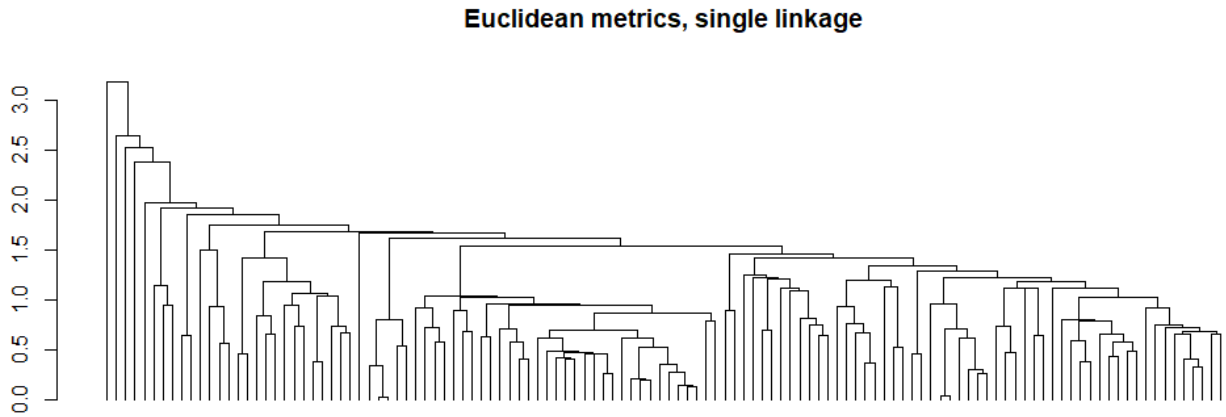




На жаль, досягти адекватного результату не вдалося. Сенсу розглядати 5 кластерів немає, адже що для трьох, що для чотирьох вже існуючі розфарбування часто перетинаються, що свідчить про невдалу кластеризацію.

Застосуємо для такої же відстані метод одного зв’язку.

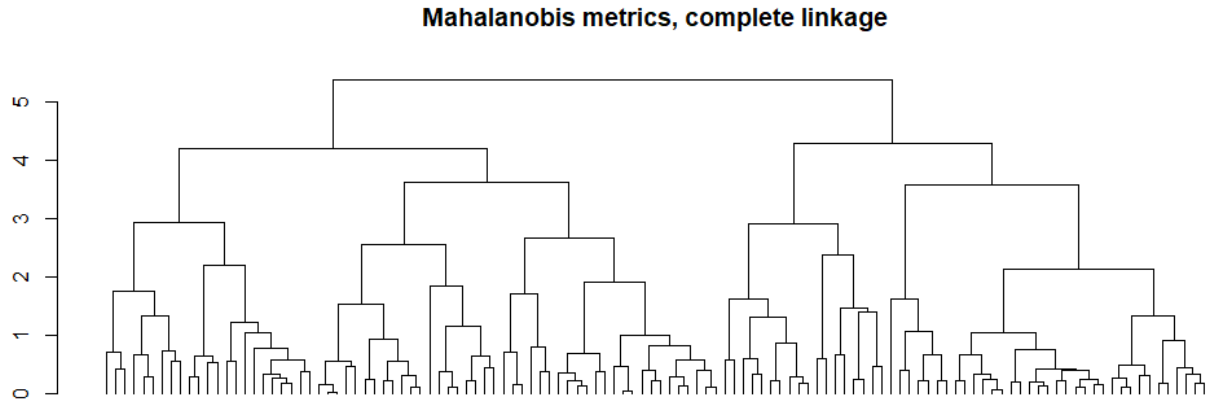
```
> # дендрограма, метод одного зв'язку
> plot(as.dendrogram(hclust(pca_d_e, method = 'single')), leaflab = 'none',
+      main = 'Euclidean metrics, single linkage')
```



Схоже на дендрограму для початкових даних. Легко бачити, що у випадку двох, трьох, та навіть п’яти кластерів – всі, крім одного будуть одноелементними (а останній – міститиме решту). Тому тут навіть немає сенсу розглядати діаграму розсіювання.

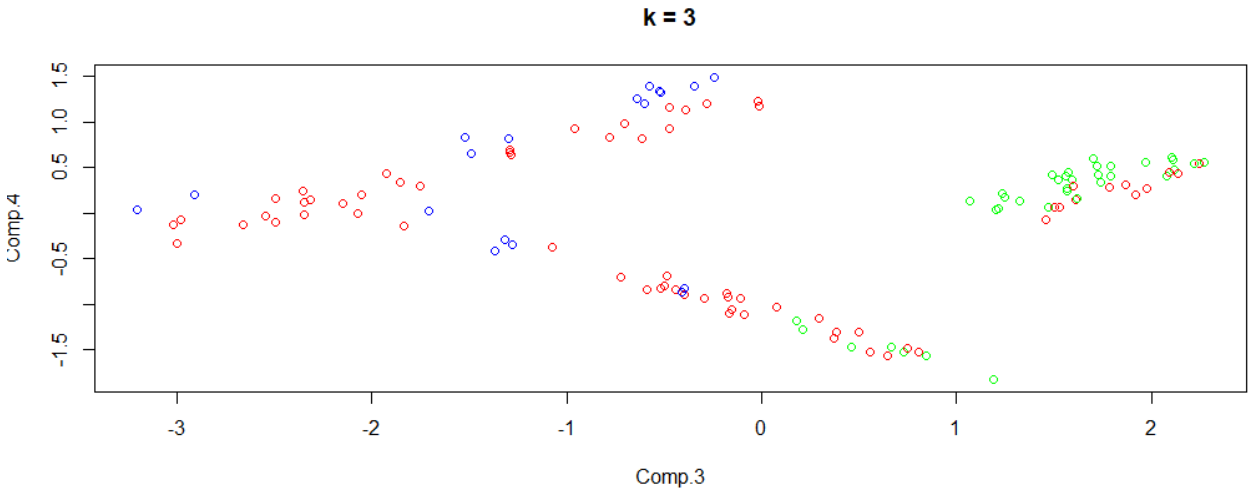
Тепер використаємо метрику Махаланобіса.

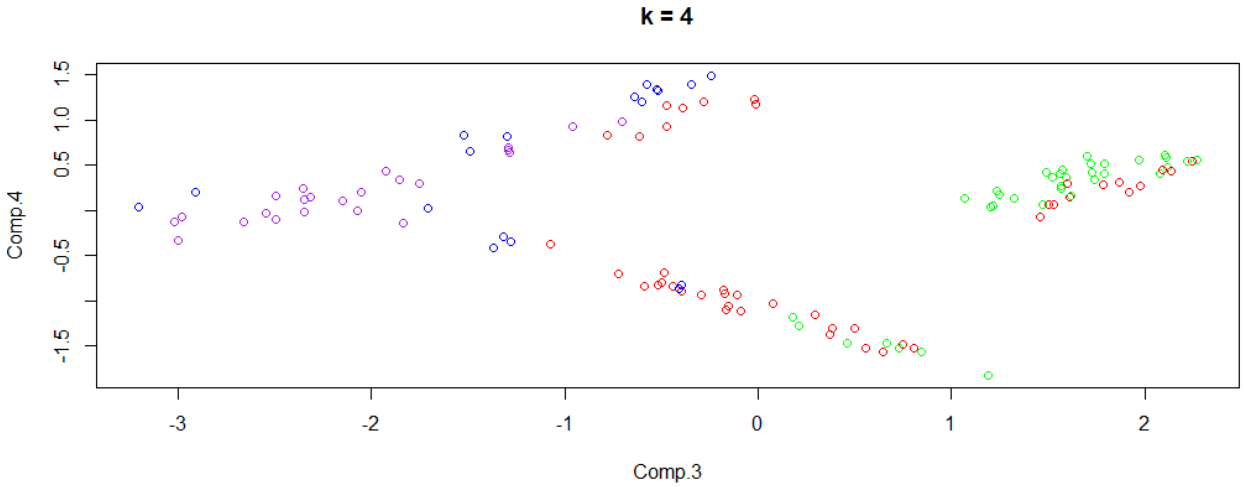
```
> library(StatMatch)
> pca_d_m <- mahalanobis.dist(res$scores[,2:4])
> pca_d_m <- as.dist(pca_d_m)
> plot(as.dendrogram(hclust(pca_d_m, method = 'complete')), leaflab = 'none',
+      main = 'Mahalanobis metrics, complete linkage')
```



Тут питання кількості кластерів ще більш відкрите: на мою думку, їх тут можна було би виділити від 2 до навіть 10.

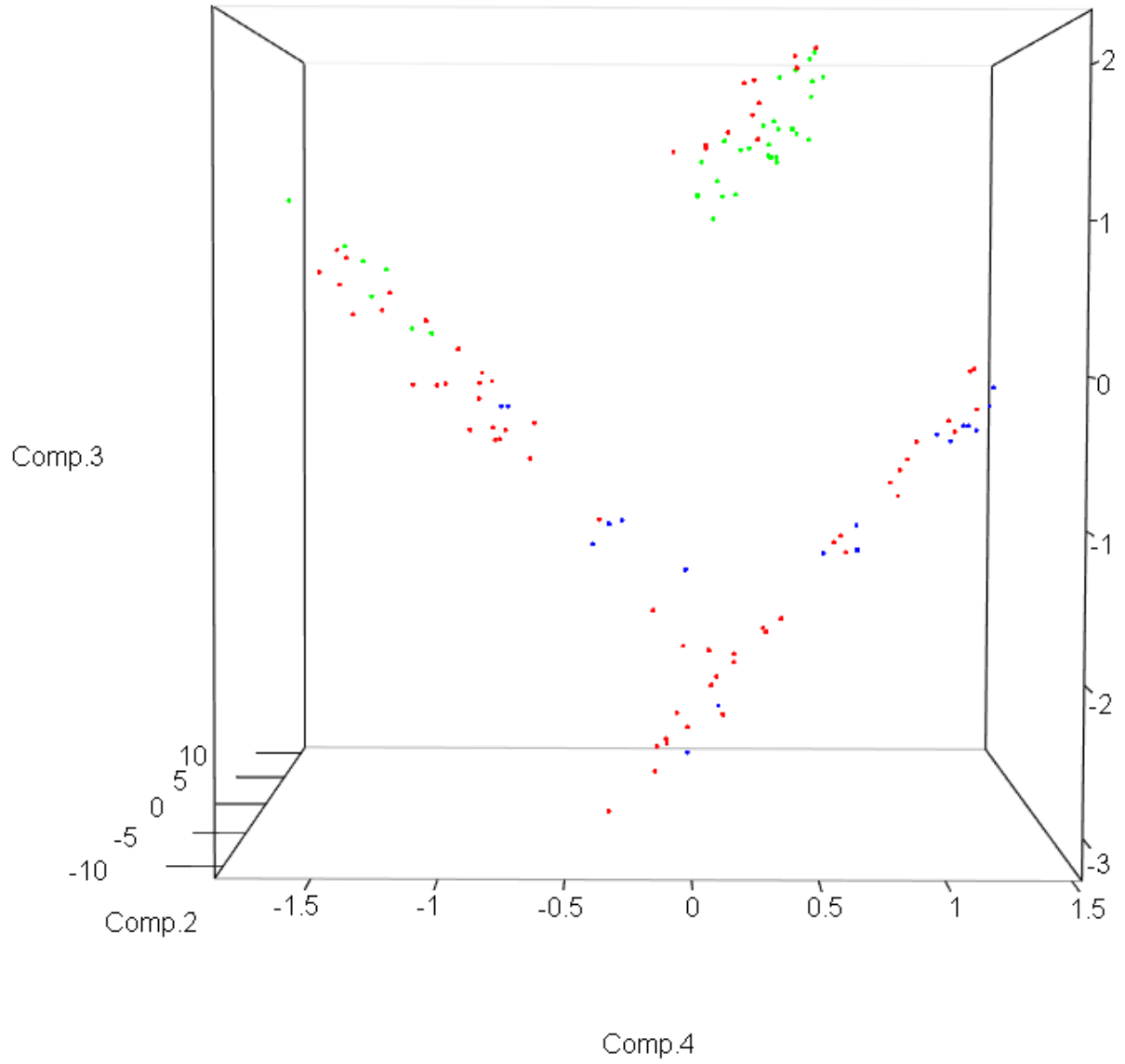
Поглянемо на початкові припущення (3 або 4).





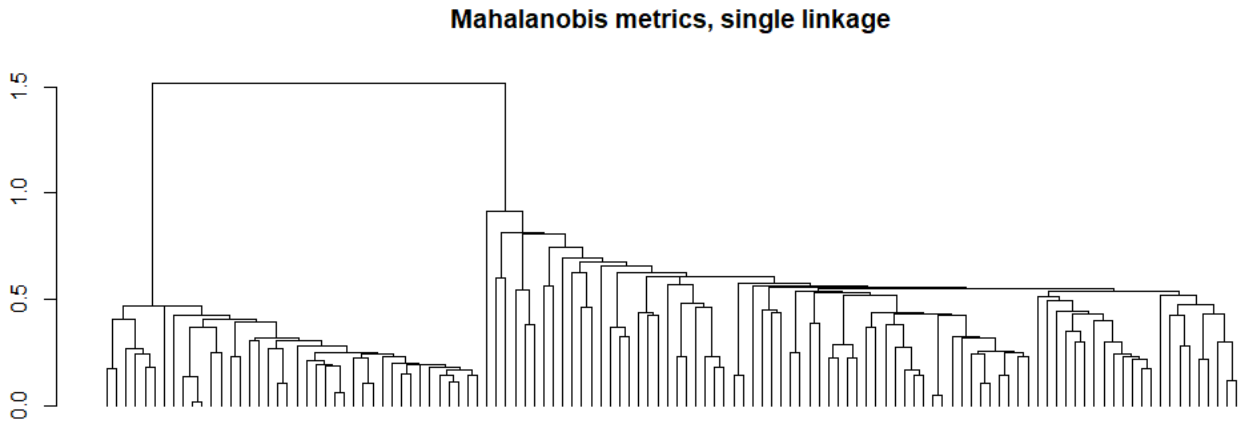
Тут, насправді, маємо найбільш адекватний результат серед усіх попередніх, але все одно не те, чого ми очікуємо. Про всяк випадок переконаємося в цьому на тривимірній діаграмі.

```
> plot3d(res$scores[,2:4], col = c('red', 'green', 'blue')[cutree(hclust(pca_
d_m, method = 'complete'), k = 3)])
```



Дійсно, не те. Тепер для даної метрики застосуємо метод найближчого сусіда (або одного зв’язку).

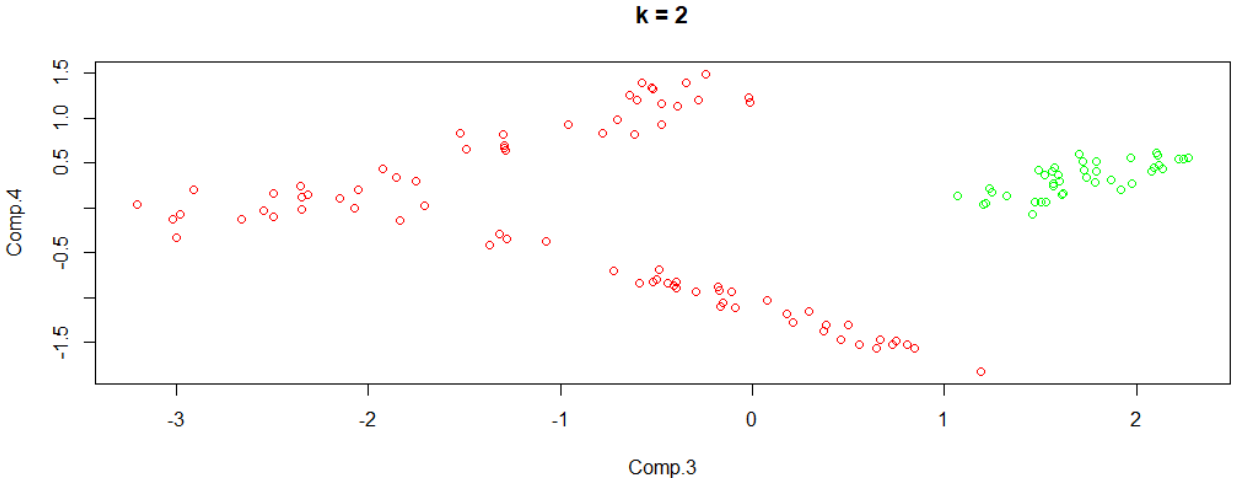
```
> # дендрограма, метод одного зв'язку
> plot(as.dendrogram(hclust(pca_d_m, method = 'single')), leaflab = 'none',
+      main = 'Mahalanobis metrics, single linkage')
```



Тут дуже і дуже чітко виділяються два кластери: на рахунок їхньої більшої кількості я би не став стверджувати.

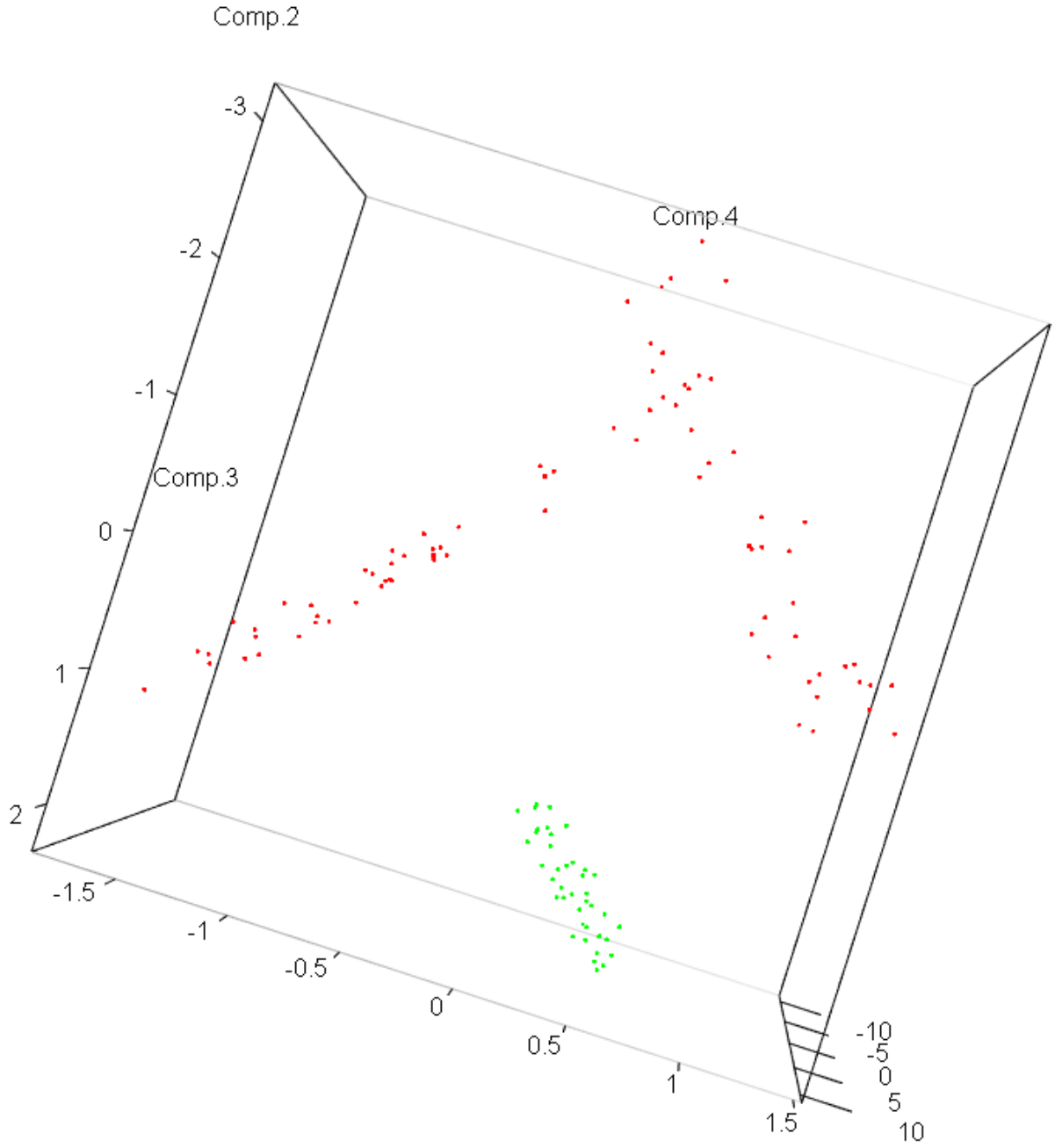
Виведемо діаграму розсіювання у просторі 3 і 4 головних компонент.

```
> # відповідна їй діаграма розсіювання у просторі 3 і 4 компонент із розфарбу
ванням на 2 кластери
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green')[cutree(hclust(pca_d_m, method = 'single'), k =
2)],
+      main = 'k = 2')
```



Отже, нарешті можемо спостерігати потрібну нам ситуацію. Поглянемо на це у тривимірній діаграмі 2-4 компонент.

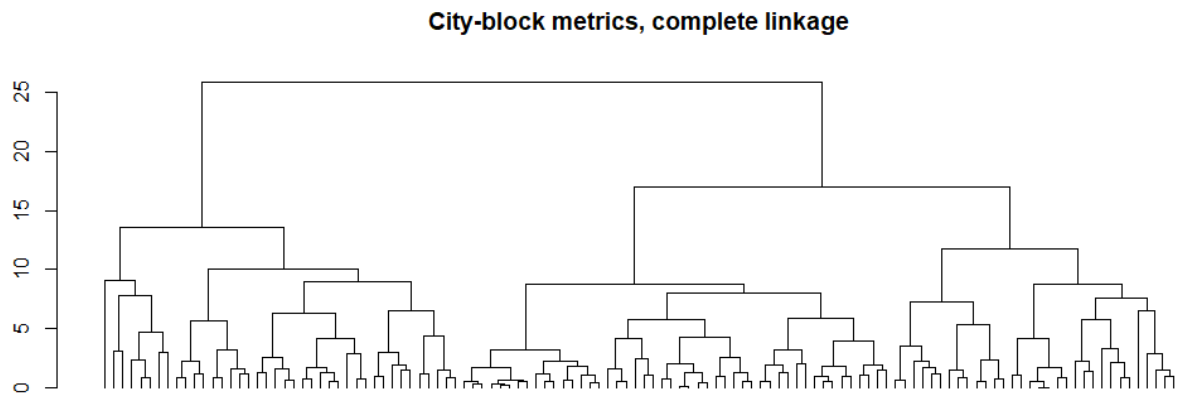
```
> # і відповідна тривимірна діаграма
> plot3d(res$scores[,2:4],
+        col = c('red', 'green')[cutree(hclust(pca_d_m, method = 'single'), k
= 2)])
```



Так, це дійсно те, на що і було очікувано.

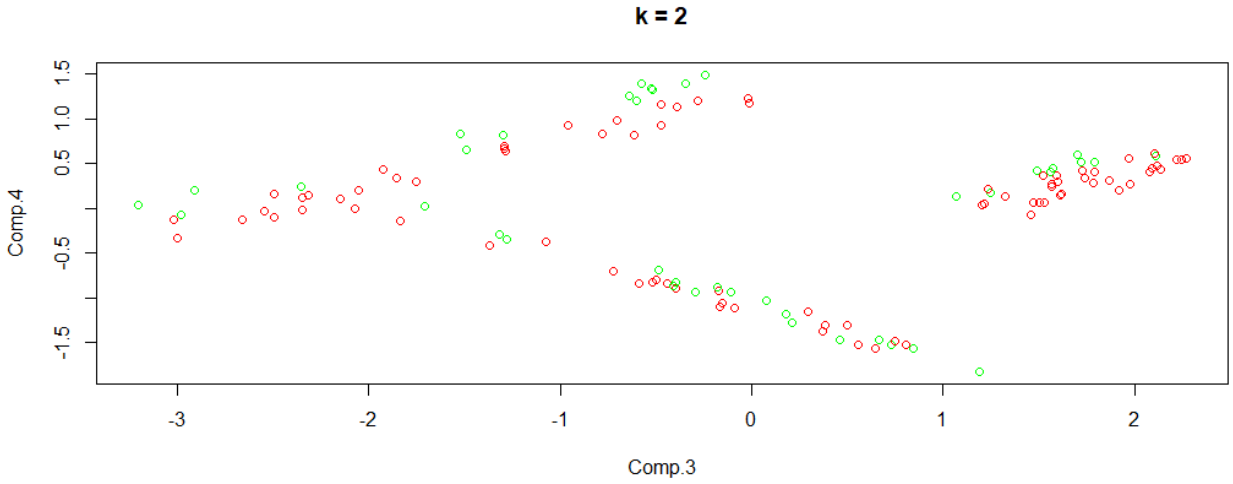
Тепер використаємо метрику сіті-блок (або манхаттанську відстань).

```
> pca_d_c <- dist(res$scores[,2:4], method = 'manhattan')
> plot(as.dendrogram(hclust(pca_d_c, method = 'complete')), leaflab = 'none',
+      main = 'City-block metrics, complete linkage')
```



Знову ж таки: чи то 2, чи то до навіть 5 кластерів… Подивимось на 2.

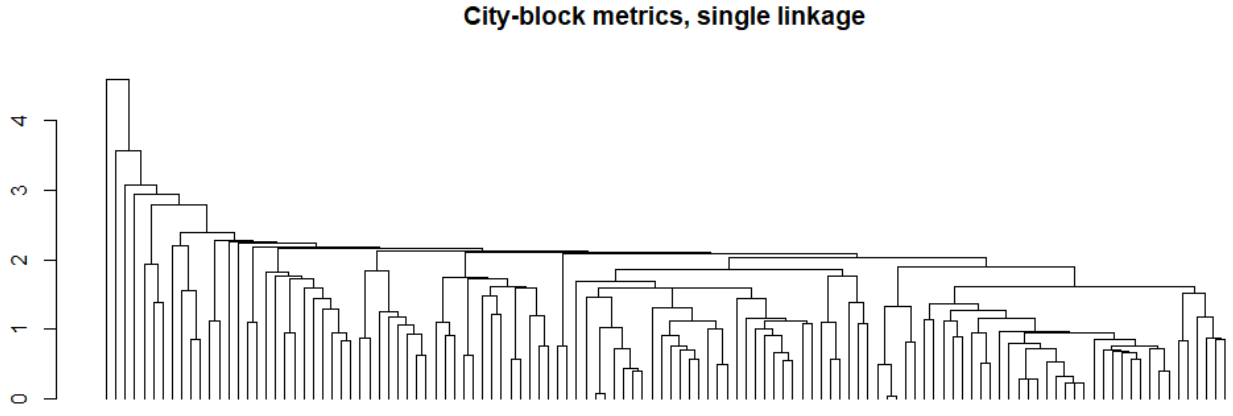
```
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green')[cutree(hclust(pca_d_c,
+                                            method = 'comp
lete'), k = 2)],
+      main = 'k = 2')
```



Оскільки кластеризація на 2 виявилась невдалою, вочевидь і решта не дадуть адекватних результатів, адже більша кількість кластерів утворюватиметься з поділу вже існуючих, які і так є невдалими.

А як щодо вже один раз спрацювавшого методу найближчого сусіда?

```
> # дендрограма, метод одного зв'язку
> plot(as.dendrogram(hclust(pca_d_c, method = 'single')), leaflab = 'none',
+      main = 'City-block metrics, single linkage')
```

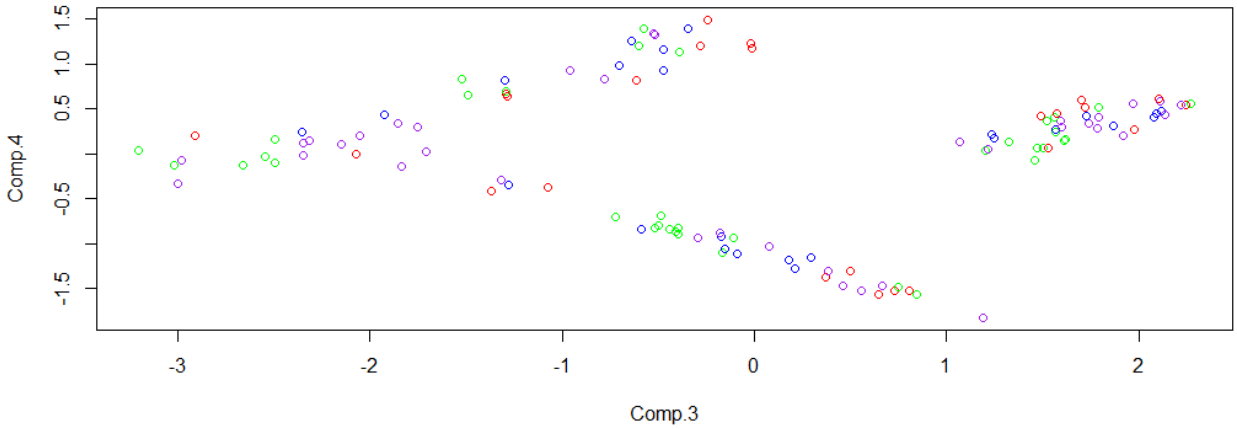


Ні, на жаль, вдруге не вистрілить. Сенсу розглядати діаграми розсіювання немає.

Що ж, звернімось тепер до техніки спектральної кластеризації.

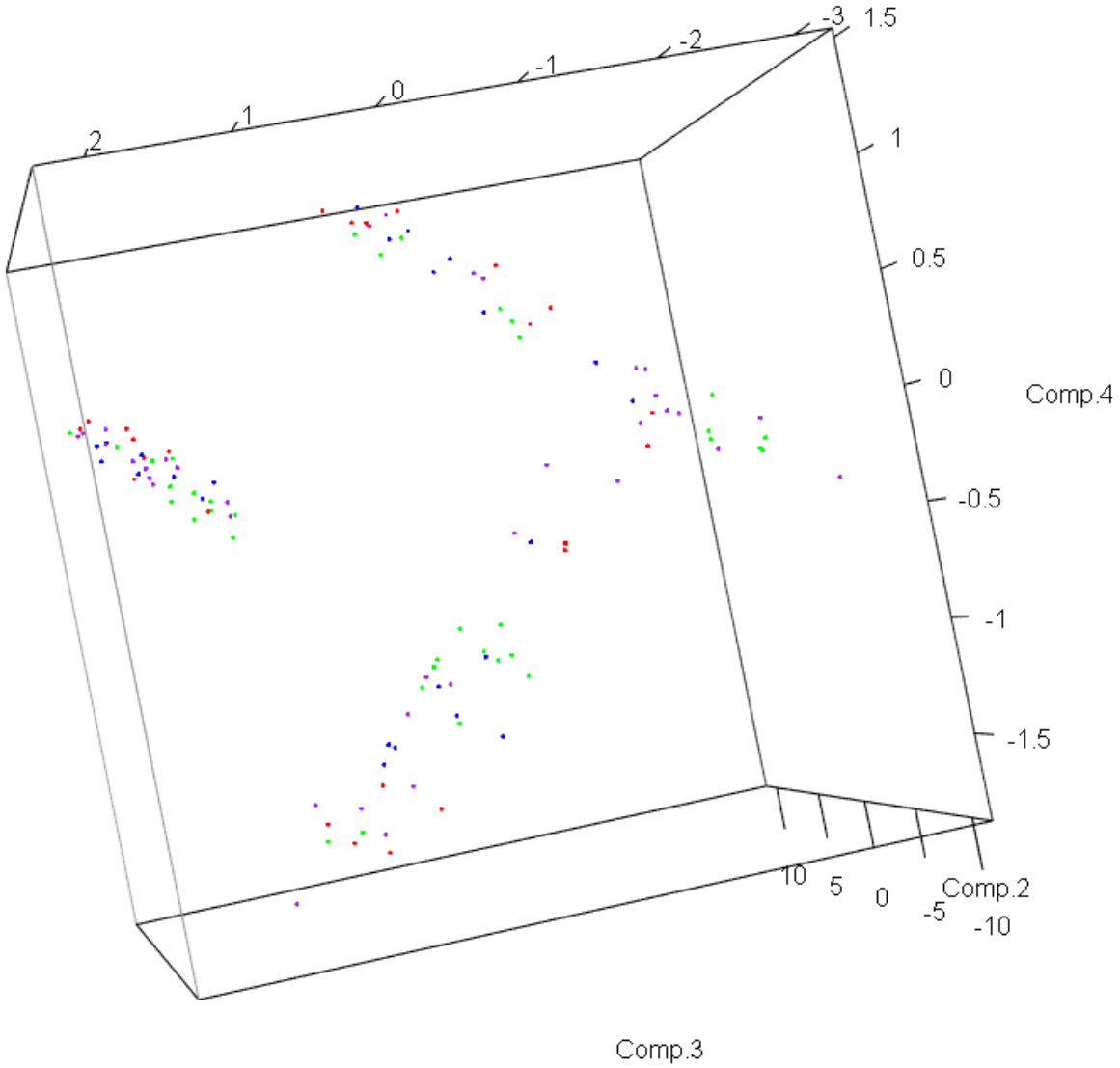
Спочатку застосуємо її до початкових даних, і вважатимемо кількість кластерів рівною 4.

```
> library(kernlab)
> sk <- specc(data, centers = 4)
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green', 'blue', 'purple')[sk])
```



Абсолютно ніякого покращення…

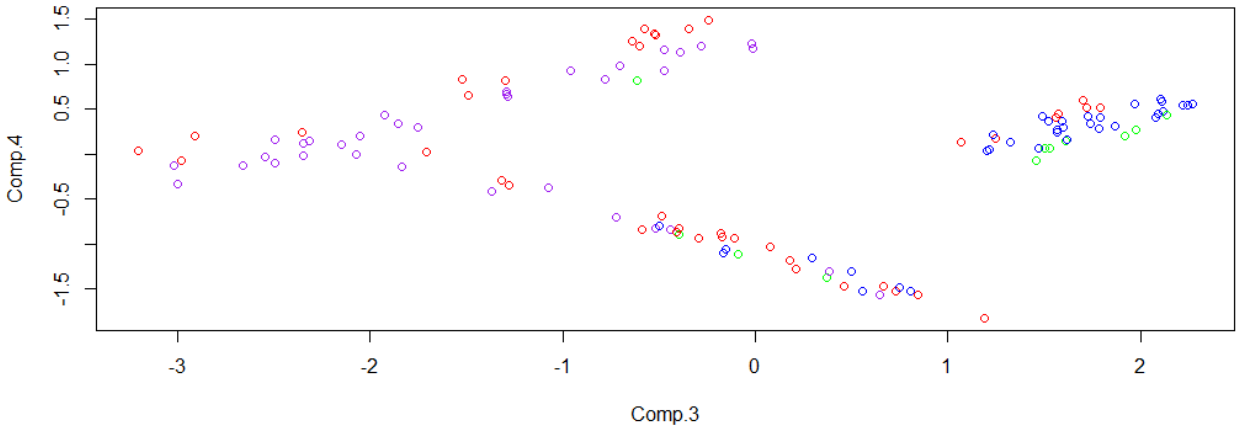
```
> plot3d(res$scores[,2:4], col = c('red', 'green', 'blue', 'purple')[sk])
```



В тому числі, і в тривимірному випадку…

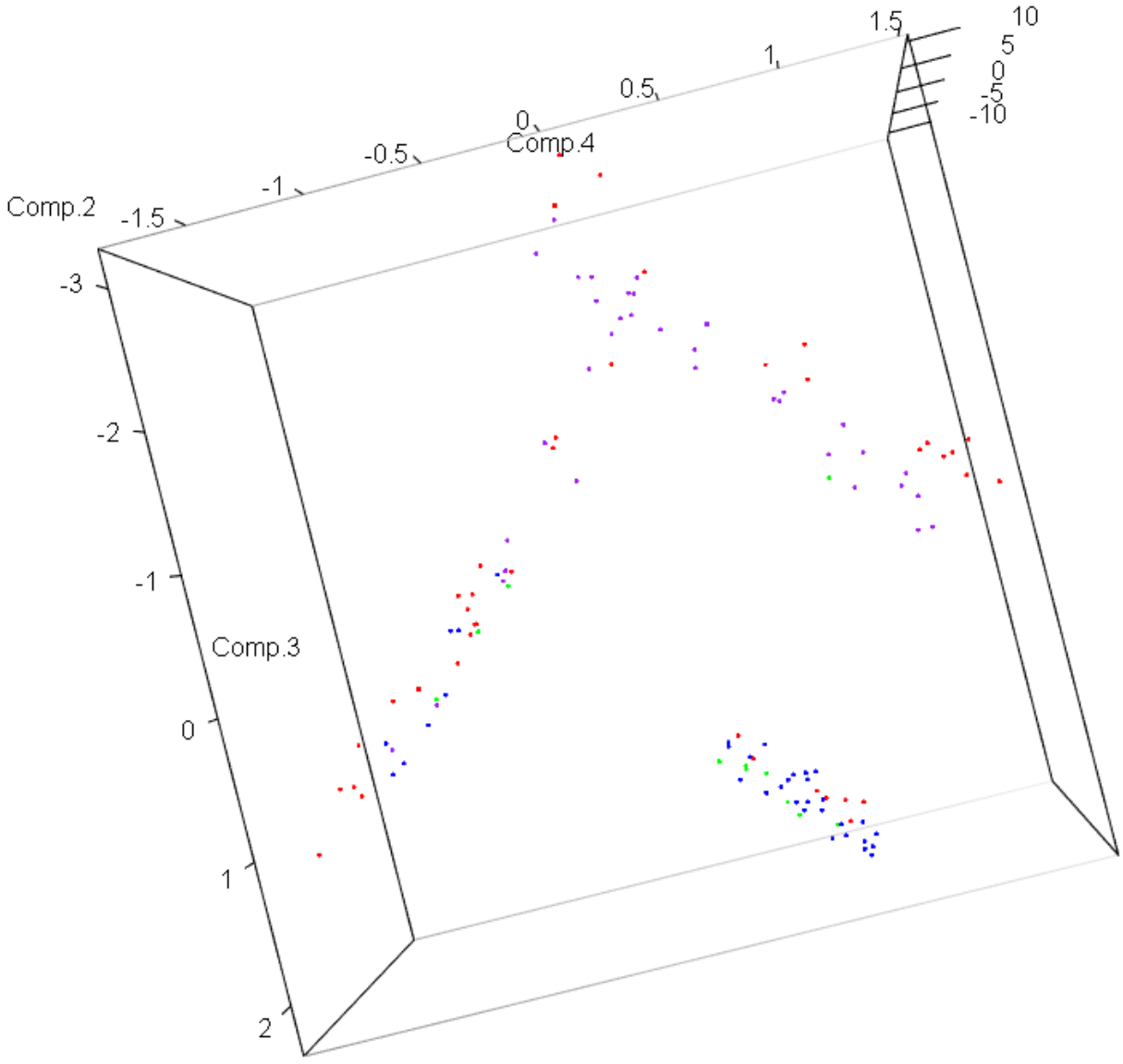
Тому спробуємо застосувати дану техніку на даних, отриманих методом головних компонент, а саме у просторі 2-4 компонент.

```
> sk1 <- specc(res$scores[,2:4], centers = 4)
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green', 'blue', 'purple')[sk1])
```



Важко сказати, що це саме те, на що ми розраховували…

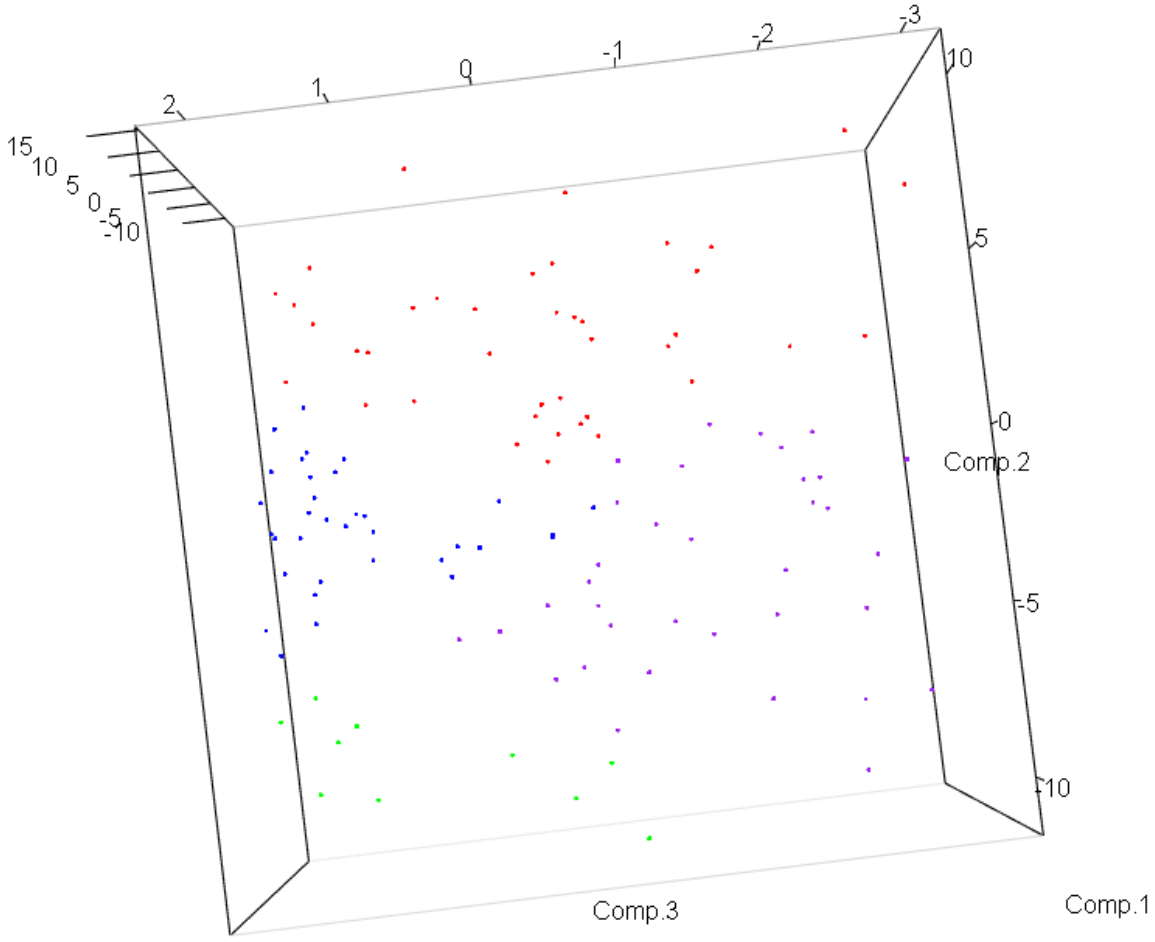
```
> plot3d(res$scores[,2:4], col = c('red', 'green', 'blue', 'purple')[sk1])
```



Це нам і підтвердить і тривимірна діаграма у просторі 2-4 компонент.

Поглянемо на тривимірну діаграму у просторі 1-3 компонент, заради інтересу.

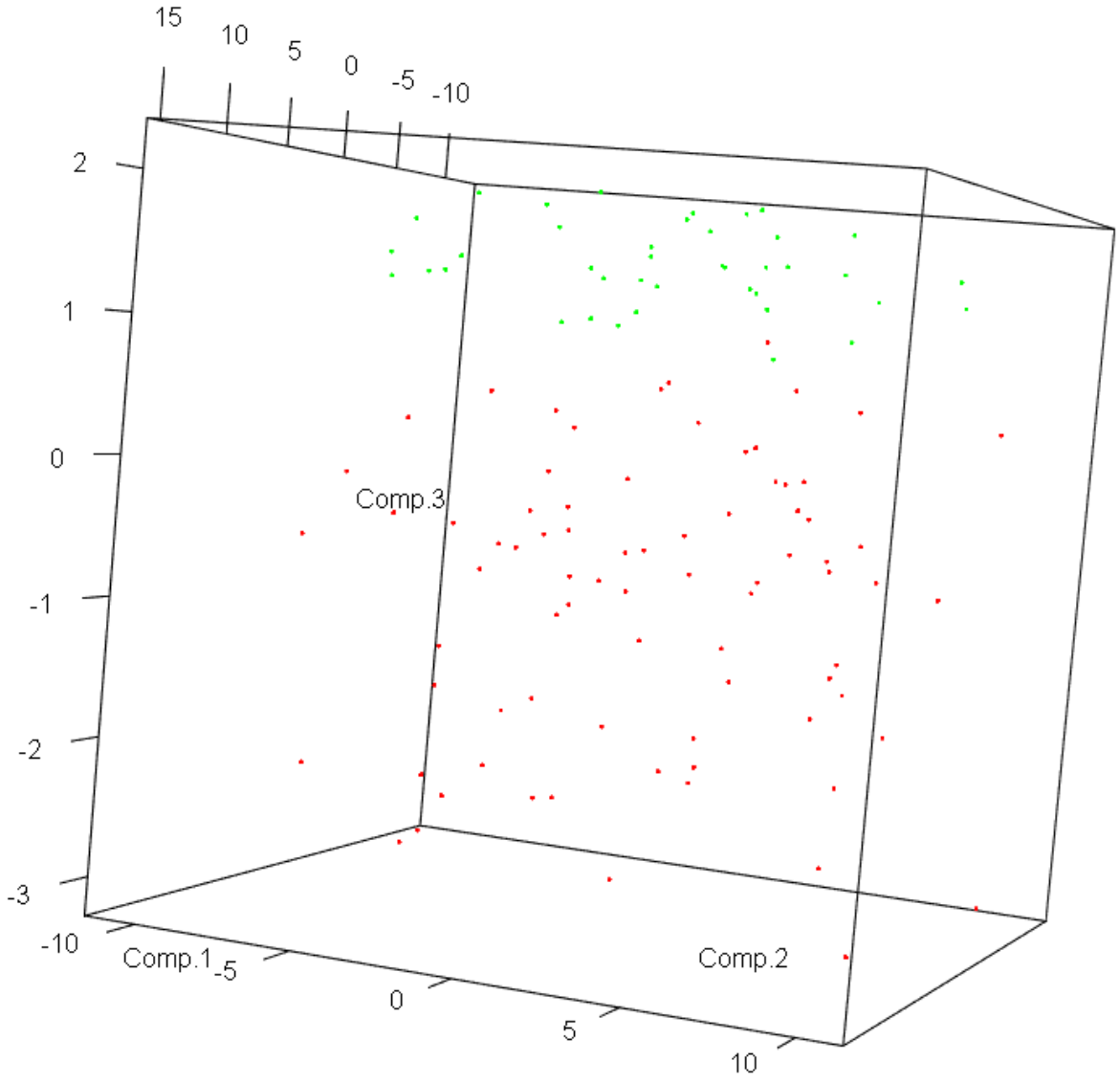
```
> plot3d(res$scores[,1:3], col = c('red', 'green', 'blue', 'purple')[sk1])
```



Так, покрутивши її деякий час, їх (кластерів), здається 4, але ж яка тут геометрична структура закладена? Як на мене, це просто купки точок, щось тут виділити дуже і дуже важко.

Отже, виходить, що спектральна кластеризація із своєю задачею не впоралась. В якості найкращої кластеризації даної роботи варто, вочевидь, обрати ієрархічну кластеризацію з методом найближчого сусіда, побудованої на відстані Махаланобіса. Поглянемо, взагалі кажучи, як виглядатиме діаграма розсіювання перших трьох головних компонент із відповідним розфарбуванням, можливо, я якось не розгледів…

```
> plot3d(res$scores[,1:3],
+        col = c('red', 'green')[cutree(hclust(pca_d_m, method = 'single'), k
= 2)])
```



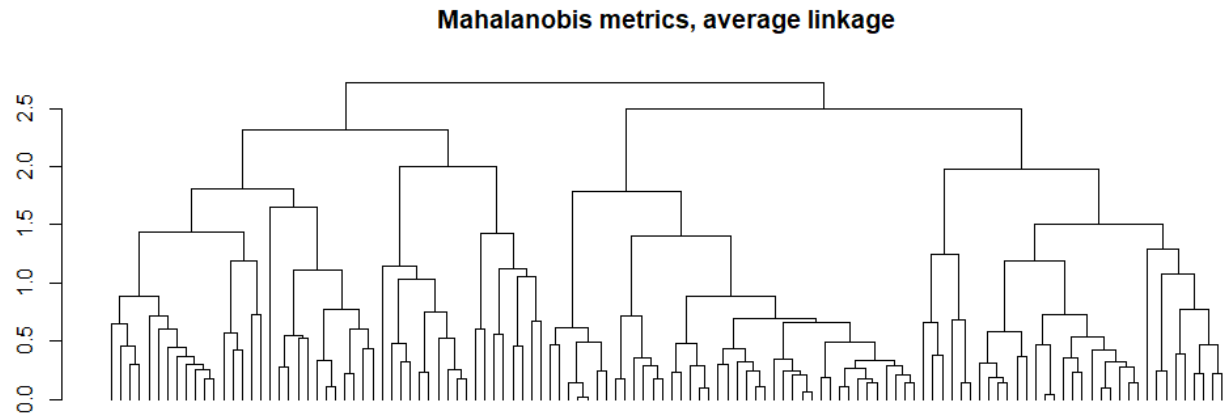
Ну, тут видно, певна структура даних, і над ними як інший кластер зелений «дашок». Але це, на мою думку, не та структура, яка легко виділяється на око.

Але ж ми явно бачимо власними очима чітко три кластери. Перепробуємо різні методи, аби хоча б який-небудь з них виділив шукані три кластери.

Експериментальним шляхом нарешті було віднайдено метод, який найближче відтворює ту структуру, яку ми, власне кажучи, і бачимо.

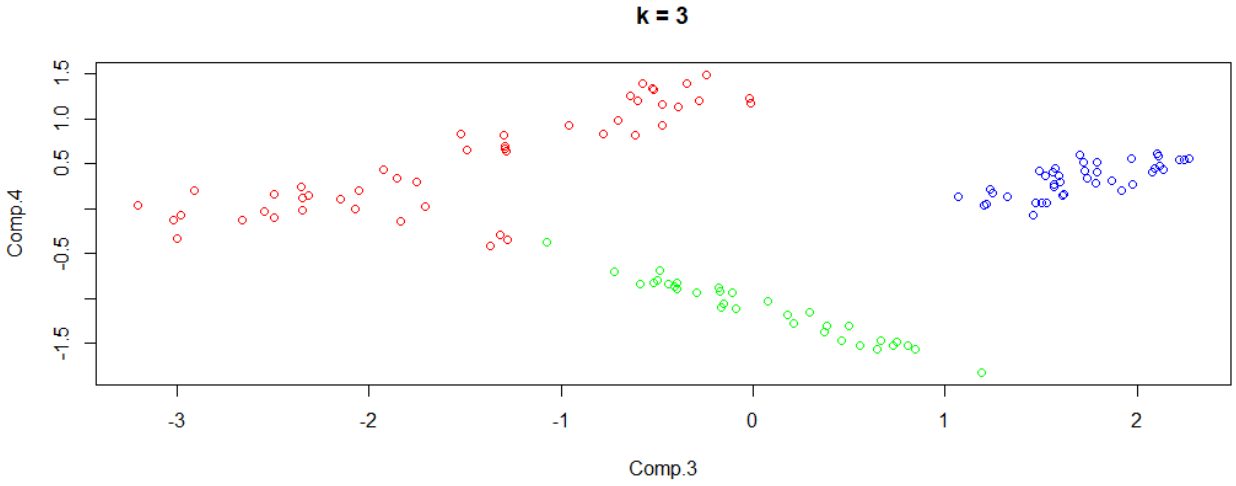
Це виявився метод середнього зв’язку для метрики Махаланобіса. Погляньмо на результати.

```
> plot(as.dendrogram(hclust(pca_d_m, method = 'average')), leaflab = 'none',
+      main = 'Mahalanobis metrics, average linkage')
```



Поділ на три кластери, як можна тут бачити, можливий. Поглянемо на діаграму розсіювання у просторі 3 і 4 головних компонент.

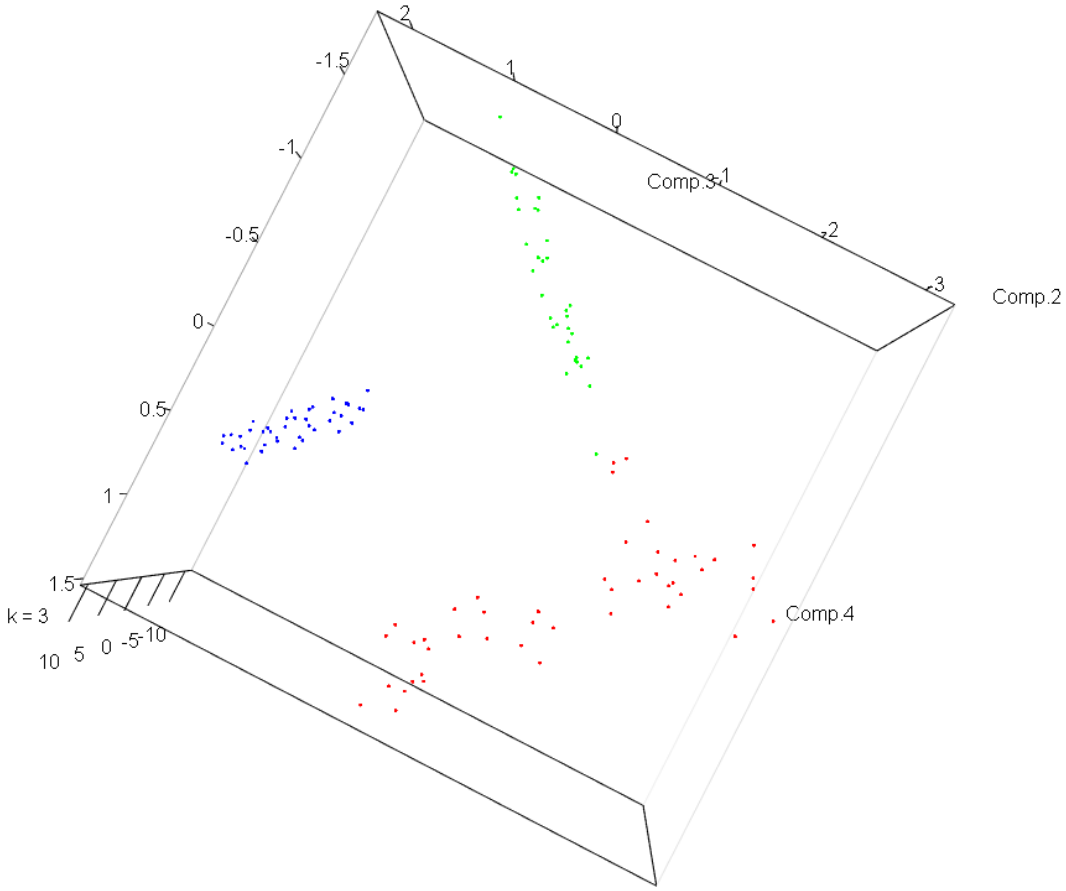
```
> plot(res$scores[,c(3,4)],
+      col = c('red', 'green','blue')[cutree(hclust(pca_d_m, method = 'averag
e'), k = 3)],
+      main = 'k = 3')
```



Здається, це саме те на що ми давно розраховували.

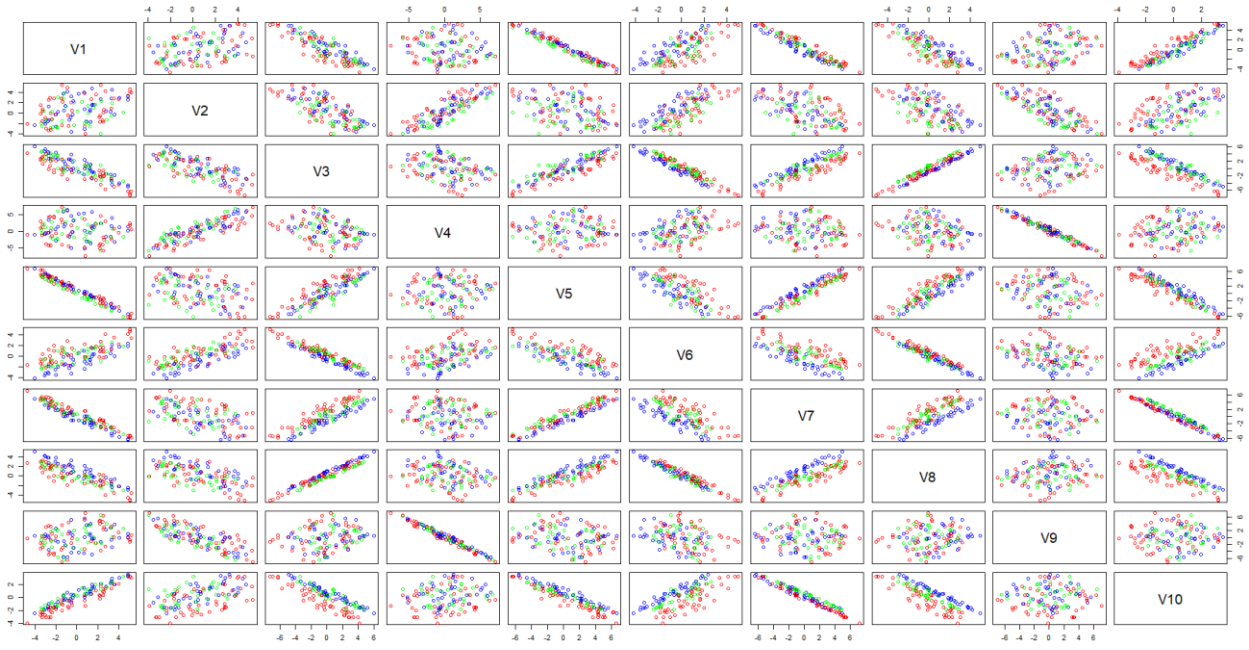
Як це вигладатиме на тій самій тривимірній діаграмі?

```
> plot3d(res$scores[,2:4], col = c('red', 'green','blue')[cutree(hclust(pca_d
_m, method = 'average'), k = 3)],
+        main = 'k = 3')
```



Так, нарешті ми отримали адекватний поділ на три кластери. Погляенемо наприкінці на попарні діаграми розсіювання із відповідним розфарбуванням вже на три кластери.

```
> pairs(data, col = c('red', 'green', 'blue')[cutree(hclust(pca_d_m, method =
'average'), k = 3)])
```



Що цікаво, так це те що така кластеризація на попарних діаграмах розсіювання в більшості випадків виходить навіть невдалою! В деяких випадках так, точки

різних кольорів розташовуються «шарами», але в деяких випадках абсолютно хаотично. Але, саме цей метод в просторі 2-4 компонент виконав адекватну кластеризацію, на відміну від решти.