ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2
«КЛАСИЧНЕ БАГАТОВИМІРНЕ ШКАЛУВАННЯ»

частина 1

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

Для даних і їх кластеризації, отриманих в обох частинах першої лабораторної роботи, застосуємо класичне багатовимірне шкалування як метод пониження розмірності.

Спершу попрацюємо **з готовими (тренувальними) даними** (тобто даними першої частини першої роботи).

Почнемо вже традиційно: з підключення бібліотек, та завантаження датасету. Також в той же самий час підгрузимо ті кластеризації, які ми обрали найкращими в першій роботі.

```
> library(factoextra)
> library(CCA)
> library(rgl)
> library(cluster)
> library(fossil) # підключаємо бібліотеки
> # зчитуємо дані
> data <- read.table('C:\\Users\\Razor\\Desktop\\дистанційне навчання\\статис
тичний аналіз багатовимірних даних\\lab1\\mult6.txt')
```

В першій частині минулої лабораторної роботи ми зупинились на трьох різних кластеризаціях – це кластеризація методом центроїдів і медоїдів для трьох кластерів (ці кластеризації ідентичні, бо для них індекс Ренда = 1), та дві кластеризації для п’яти кластерів (там індекс Ренда виходив порядку 0.94).

Перед цим не забудемо задати палітру кольорів для розфарбування.

```
> # задаємо палітру кольорів
> col = c('black', 'red', 'green', 'blue', 'orange',
+         'purple', 'yellow', 'brown', 'burlywood',
+         'deepskyblue', 'darkseagreen', 'deeppink',
+         'salmon', 'turquoise1', 'darkblue', 'darkred',
+         'aquamarine', 'grey', 'chocolate', 'magenta')
```

Реалізуємо ці три кластеризації.

```
> # метод центроїдів, к = 3
> km.res3 <- kmeans(data, 3, nstart = 25)
> # метод центроїдів, к = 5
> km.res5 <- kmeans(data, 5, nstart = 25)
> # метод медоїдів, к = 5
> pam.res5 <- pam(data, 5)
```

Тепер для наших даних виконаємо їх перетворення у таблицю відмінностей, використовуючи для цього три варіанти відстаней: евклідову, манхаттанську і максимальну.

```
> # перетворимо дані у таблицю відмінностей
> d_eucl <- dist(data, 'euclidean')
> d_mannh <- dist(data, 'manhattan')
> d_max <- dist(data, 'maximum')
```

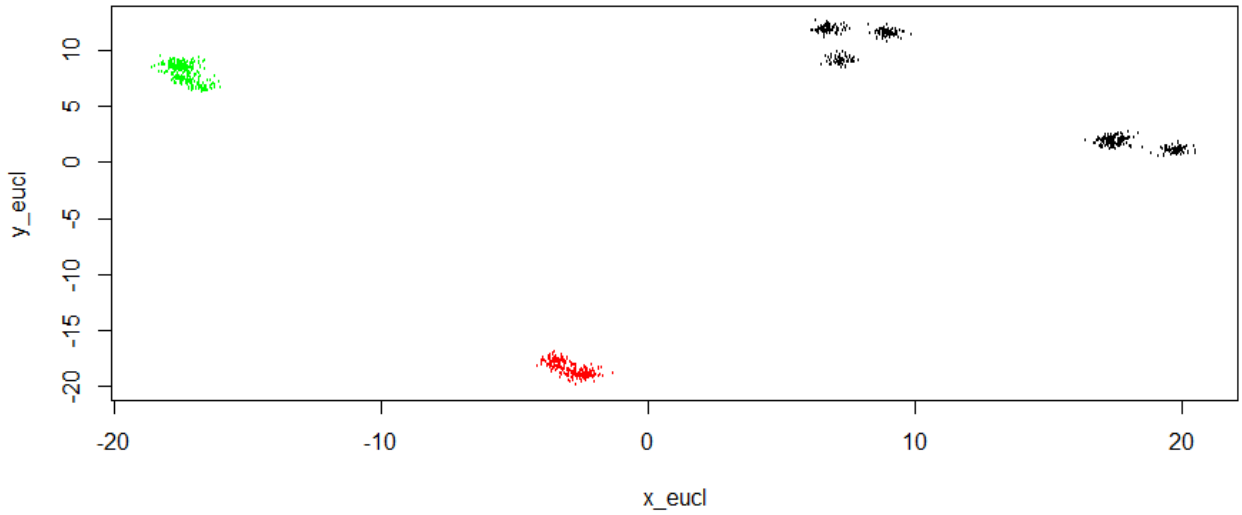
Далі за допомогою функції *cmdscale* понизимо розмірність даних, поданих у вигляді таблиці відмінностей до розмірності $k = 2$.

```
> # понизимо розмірність
> fit_eucl <- cmdscale(d_eucl, eig = TRUE, k = 2)
> fit_mannh <- cmdscale(d_mannh, eig = TRUE, k = 2)
> fit_max <- cmdscale(d_max, eig = TRUE, k = 2)
```

Далі виведемо відповідно двовимірні діаграми розсіювання для результатів, із урахуванням наших кластеризацій.

Спершу подивимось на результат, отриманий з евклідової відстані.

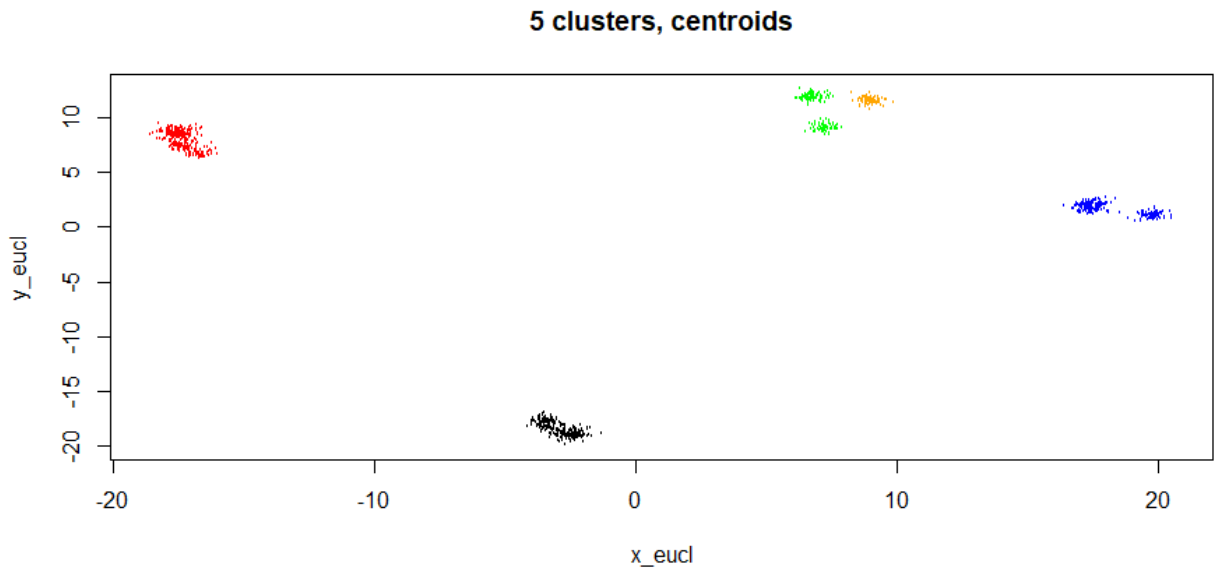
```
> # евклідова відстань
> x_eucl <- fit_eucl$points[,1]
> y_eucl <- fit_eucl$points[,2]
> # k = 3
> plot(x_eucl, y_eucl, col = col[km.res3$cluster], cex = 0.3)
```

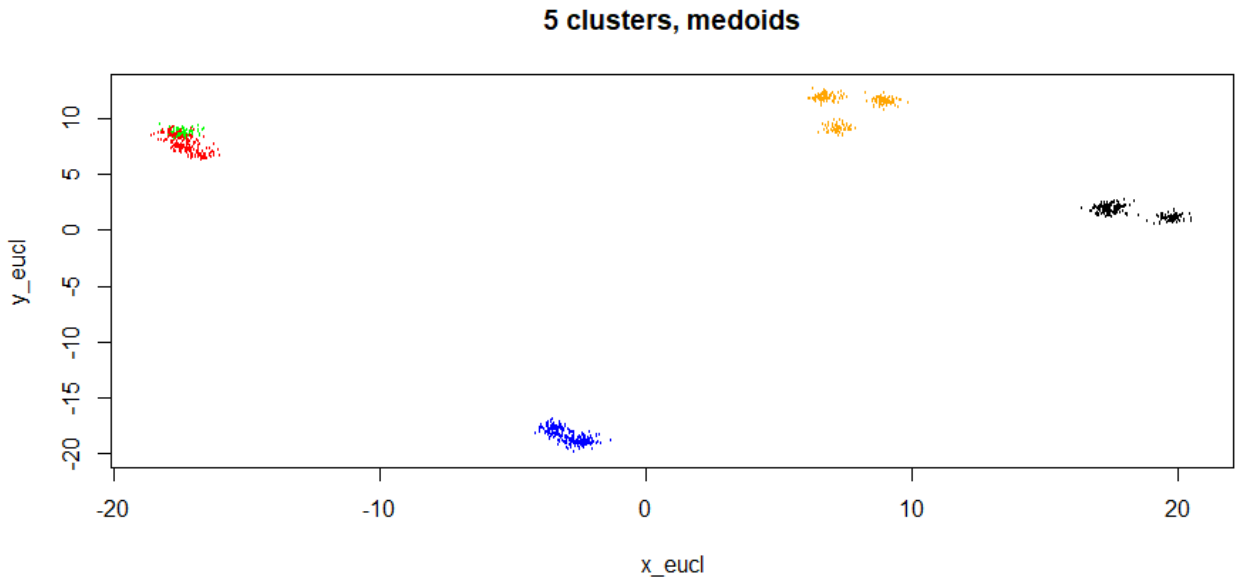


З даної діаграми можемо бачити, що тут варто стверджується більше на користь 4 кластерів, аніж 3. З іншої сторони, групка спостережень зверху є в певній мірі об’єднанням трьох маленьких групок.

Розфарбуємо для 5 кластерів.

```
> # k = 5, centroids
> plot(x_eucl, y_eucl, col = col[km.res5$cluster], cex = 0.3, main = '5 clust
ers, centroids')
> # k = 5, medoids
> plot(x_eucl, y_eucl, col = col[pam.res5$cluster], cex = 0.3, main = '5 clus
ters, medoids')
```





Розфарбування для 5 кластерів методом центроїдів картину дещо покращило, але саме в тій групі з трьох маленьких купок у нас одна купка – одного кольору, дві інші – іншого.

Схожа ситуація і в методі медоїдів, хоча тут всі маленькі групки зібрано разом. Тепер недолік в тому, що в групці зліва присутня своя групка (в червоних точках зосереджені зелені).

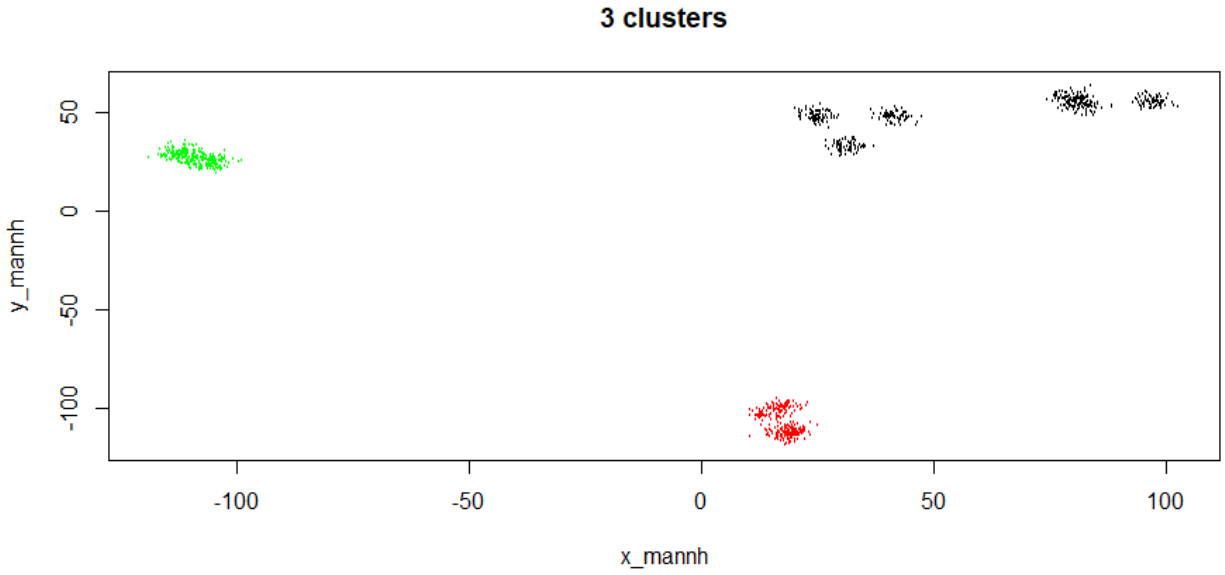
Тобто, на евклідових відстанях, є підстави стверджувати на користь 4 кластерів… Що цікаво, жодна з діаграм в першій роботі на користь такого не свідчили.

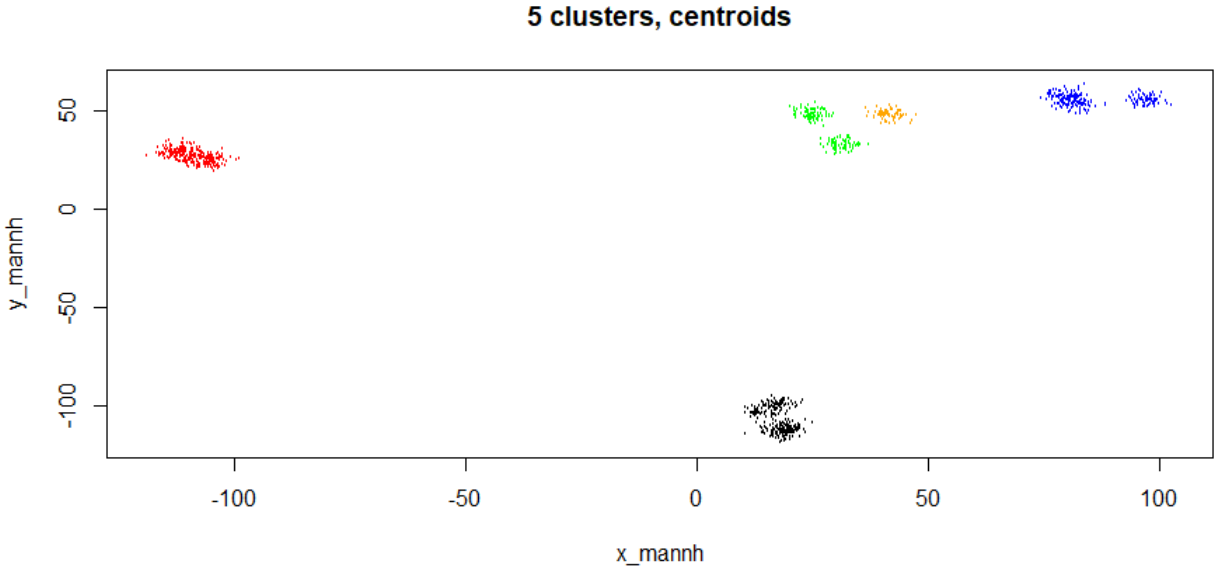
Спробуємо далі манхаттанську відстань.

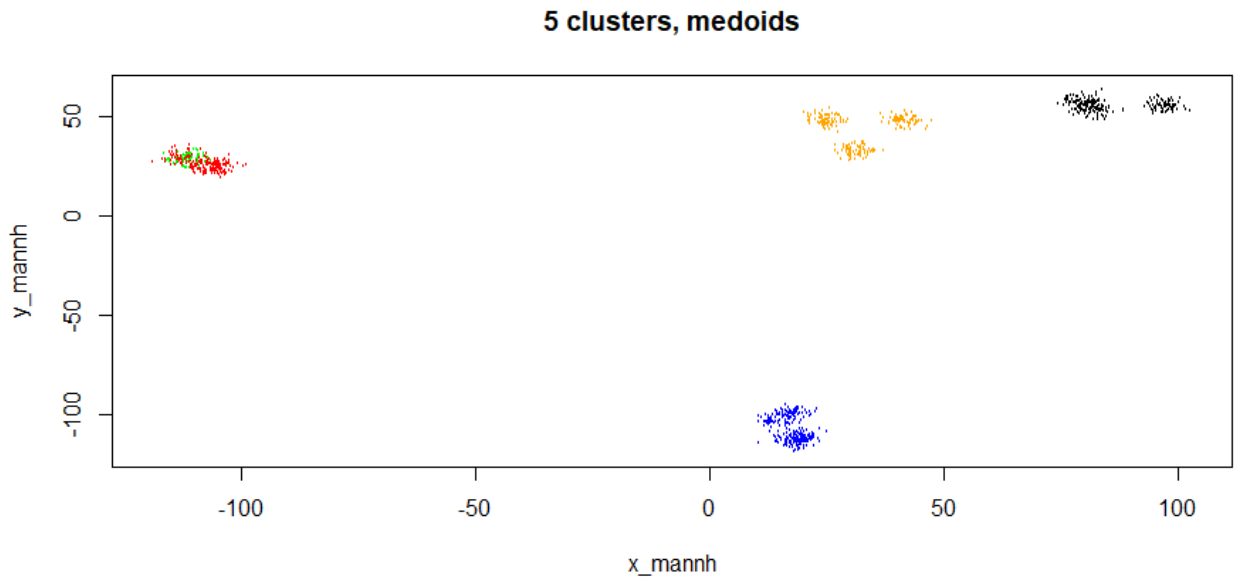
```
> # манхаттанська відстань
> x_mannh <- fit_mannh$points[,1]
> y_mannh <- fit_mannh$points[,2]
> # k = 3
```

```
> plot(x_mannh, y_mannh, col = col[km.res3$cluster], cex = 0.3, main = '3 clu
sters')
> # k = 5, centroids
> plot(x_mannh, y_mannh, col = col[km.res5$cluster], cex = 0.3, main = '5 clu
sters, centroids')
> # k = 5, medoids
> plot(x_mannh, y_mannh, col = col[pam.res5$cluster], cex = 0.3, main = '5 cl
usters, medoids')
```

Резултати нижче.





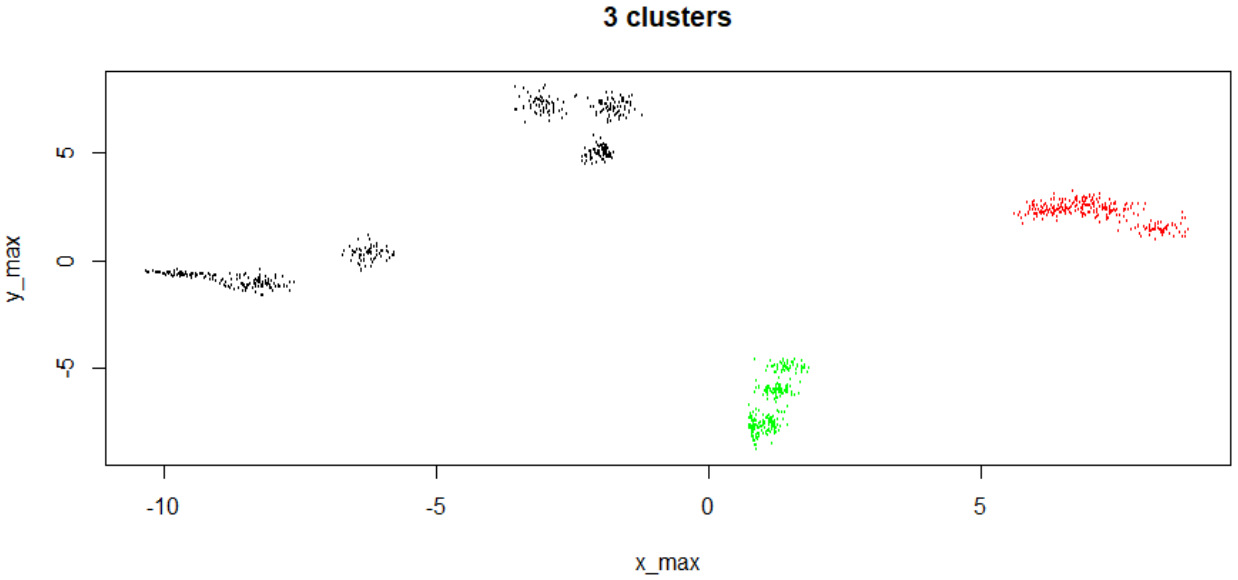


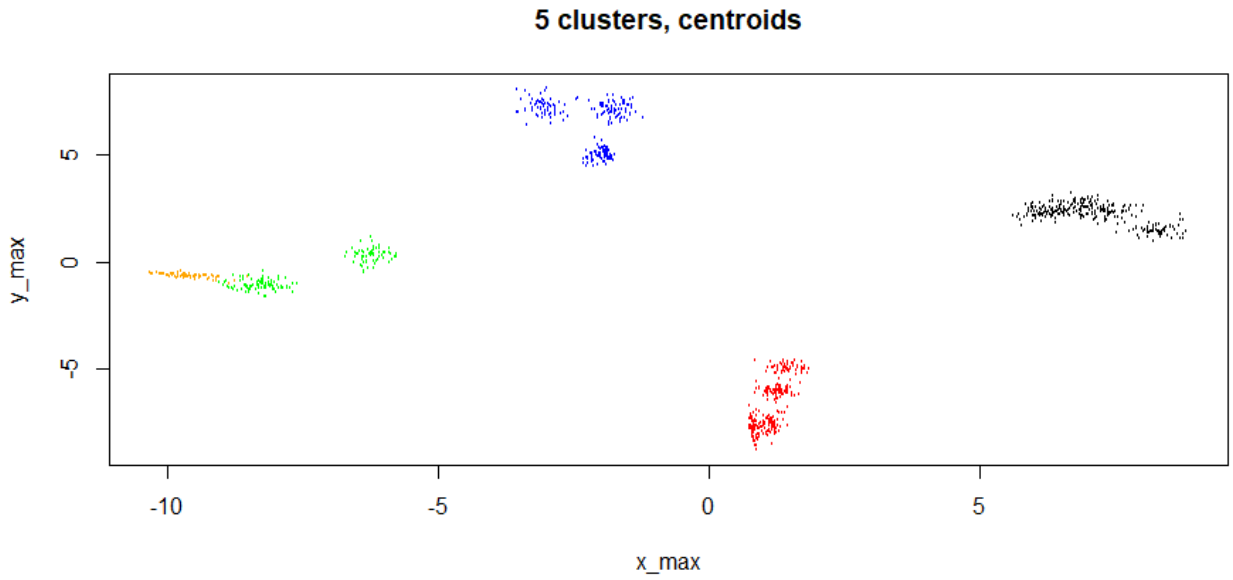
Можемо бачити, що, насправді, принципових відмінностей від еквлідової відстані виявити не представляється можливим – поведінка у всіх випадках аналогічна.

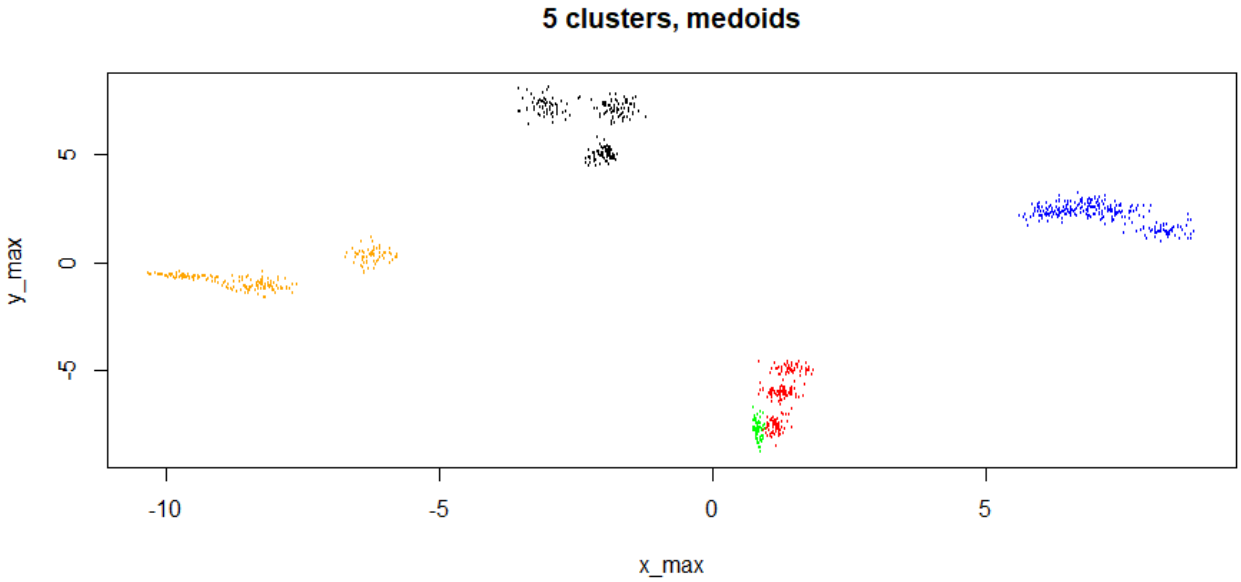
І наприкінці застосуємо максимальну відстань.

```
> # максимальна відстань
> x_max <- fit_max$points[,1]
> y_max <- fit_max$points[,2]
> # k = 3
> plot(x_max, y_max, col = col[km.res3$cluster], cex = 0.3, main = '3 cluster
s')
> # k = 5, centroids
> plot(x_max, y_max, col = col[km.res5$cluster], cex = 0.3, main = '5 cluster
s, centroids')
> # k = 5, medoids
> plot(x_max, y_max, col = col[pam.res5$cluster], cex = 0.3, main = '5 cluste
rs, medoids')
```

Відповідні діаграми наведемо нижче.







Тепер точки, а точніше кажучи, групки, розташовуються у дещо інших місцях, але, в цілому, поведінка зберігається: у випадку трьох кластерів є один великий суцільний кластер, який, очевидно, розбивається мінімум на два, і як максимум, на п’ять. У випадку п’яти кластерів методом центроїдів знову ж таки маємо, здається, гарний поділ, але в лівій частині серед зелених спостережень виділяються окремо помаранчеві. І аналогічна ситуація з 5 кластерами методом медоїдів: в одній групі (знизу) виділяється всередині ще одна, що є зовсім нелогічним.

Але що можна стверджувати майже напевно, так це те, що такий метод пониження розмірності, як класичне багатовимірне шкалування, на «тренувальних» даних стверджує більше на користь 4 кластерів, аніж 3 чи 5. Навіть якщо не 4, то порядку 7-9, дивлячись з якого боку поглянути.

Тепер попрацюємо **з практичними даними** (тобто даними другої частини 1 роботи).

Як мені підказали колеги в групі, нормувати та центрувати не обов'язково вручну, це все робиться однією командою.

```
> # зчитуємо дані
> data1 <- read_excel('data.xlsx')
> rows <- t(data1[,1])
> data1 <- data1[,-1]
>
> # центрування і нормування
>
> data1 <- as.data.frame(scale(data1))
> row.names(data1) <- rows
```

Нагадаємо, що в другій частині першої роботи ми зупинилися на варіанті $k = 4$ отриманим методом медоїдів. Проте, на мою думку, можна було би ще розглянути такі варіанти, як $k = 9$, отримані як методом медоїдів, так і центроїдів.

```
> # метод медоїдів, k = 4
> pam.res41 <- pam(data1, 4)
>
> # метод медоїдів, k = 9
> pam.res91 <- pam(data1, 9)
>
> # метод центроїдів, k = 9
> km.res91 <- kmeans(data1, 9, nstart = 25)
```

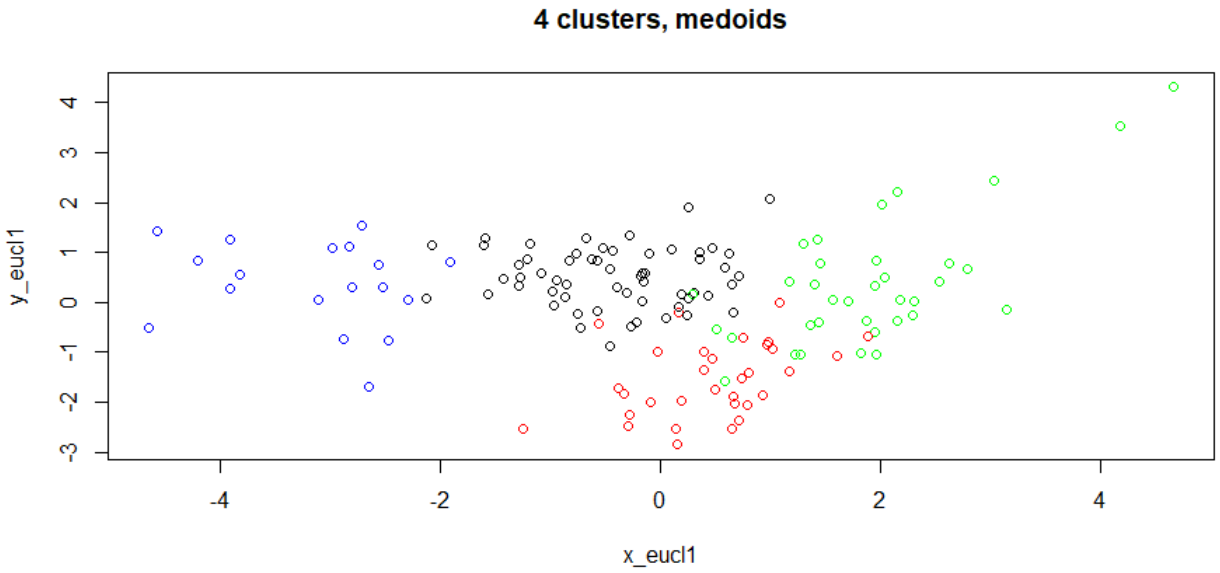
Далі проробимо абсолютно аналогічні дії, як і з даними першої частини: перетворимо дані у таблиці відмінностей, і понизимо розмірність.

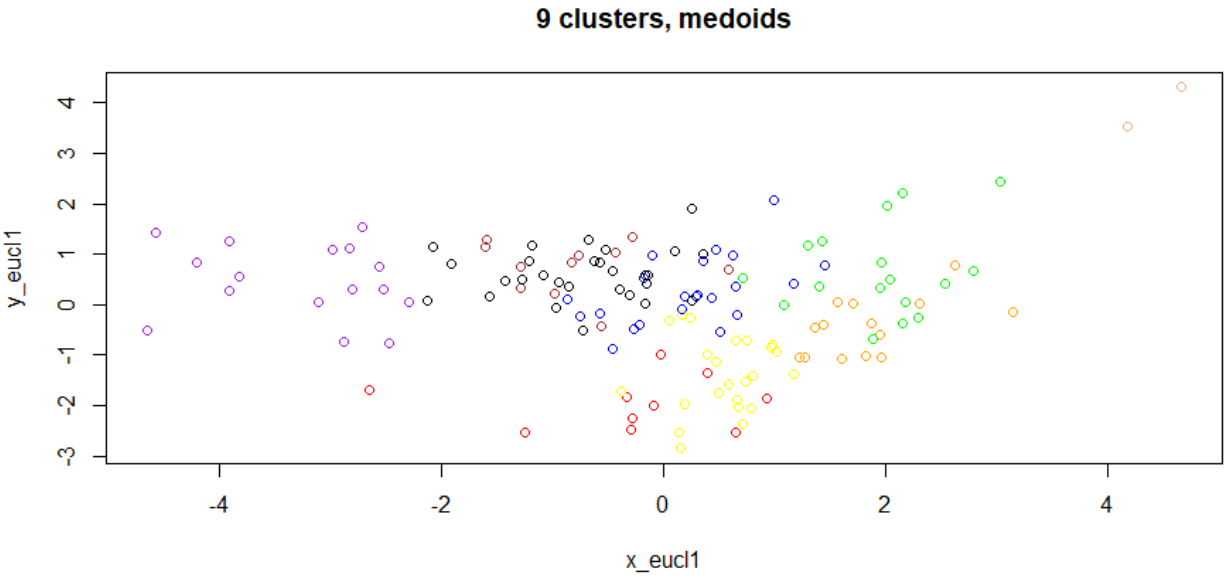
```
> # перетворимо дані у таблицю відмінностей
> d_eucl1 <- dist(data1, 'euclidean')
> d_mannh1 <- dist(data1, 'manhattan')
> d_max1 <- dist(data1, 'maximum')
> # понизимо розмірність
> fit_eucl1 <- cmdscale(d_eucl1, eig = TRUE, k = 2)
> fit_mannh1 <- cmdscale(d_mannh1, eig = TRUE, k = 2)
> fit_max1 <- cmdscale(d_max1, eig = TRUE, k = 2)
```

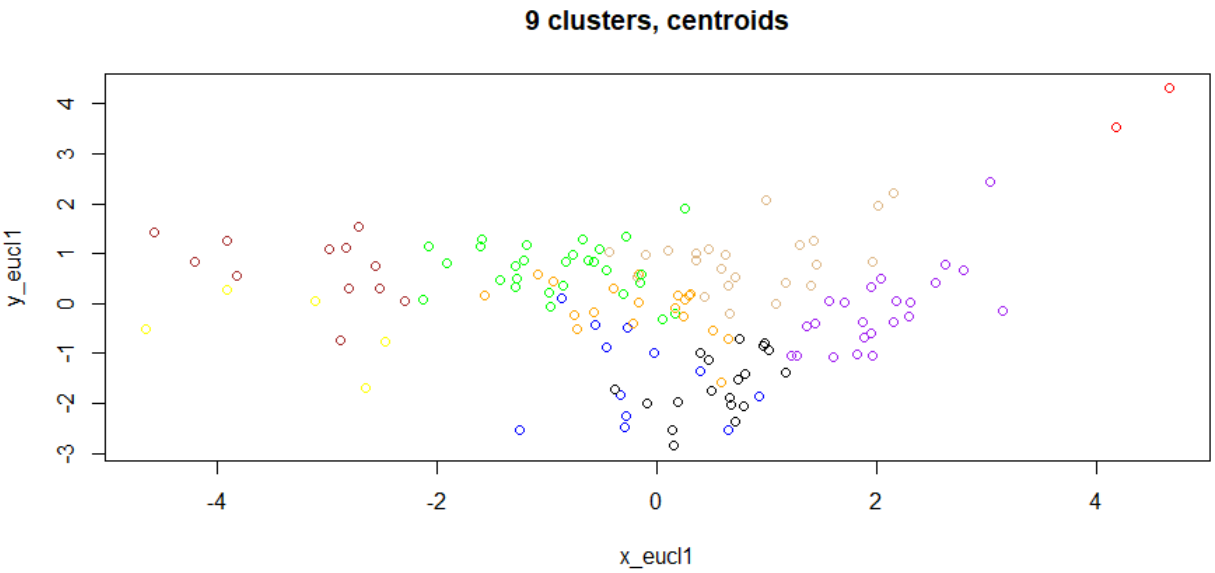
Далі виведемо відповідні діаграми розсіювання, з відповідними розфарбуваннями на кластери, отриманими в першій роботі.

Для евклідової відстані:

```
> # евклідова відстань
> x_eucl1 <- fit_eucl1$points[,1]
> y_eucl1 <- fit_eucl1$points[,2]
> # k = 4, medoids
> plot(x_eucl1, y_eucl1, col = col[pam.res41$cluster], cex = 1, main = '4 clu
sters, medoids')
> # k = 9, medoids
> plot(x_eucl1, y_eucl1, col = col[pam.res91$cluster], cex = 1, main = '9 clu
sters, medoids')
> # k = 9, centroids
> plot(x_eucl1, y_eucl1, col = col[km.res91$cluster], cex = 1, main = '9 clus
ters, centroids')
```





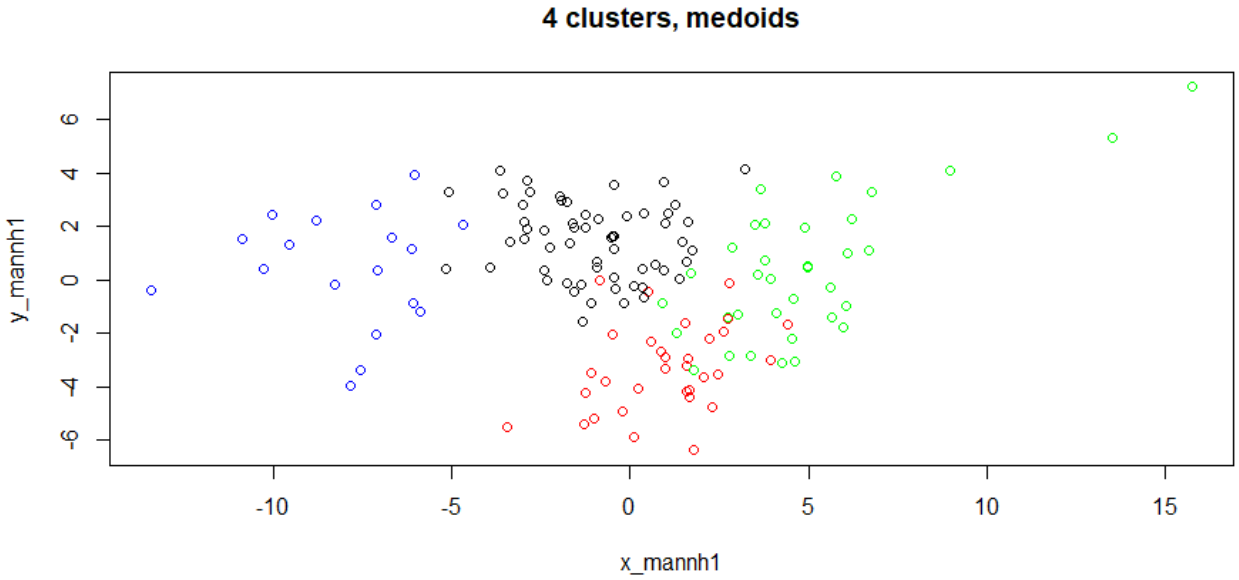


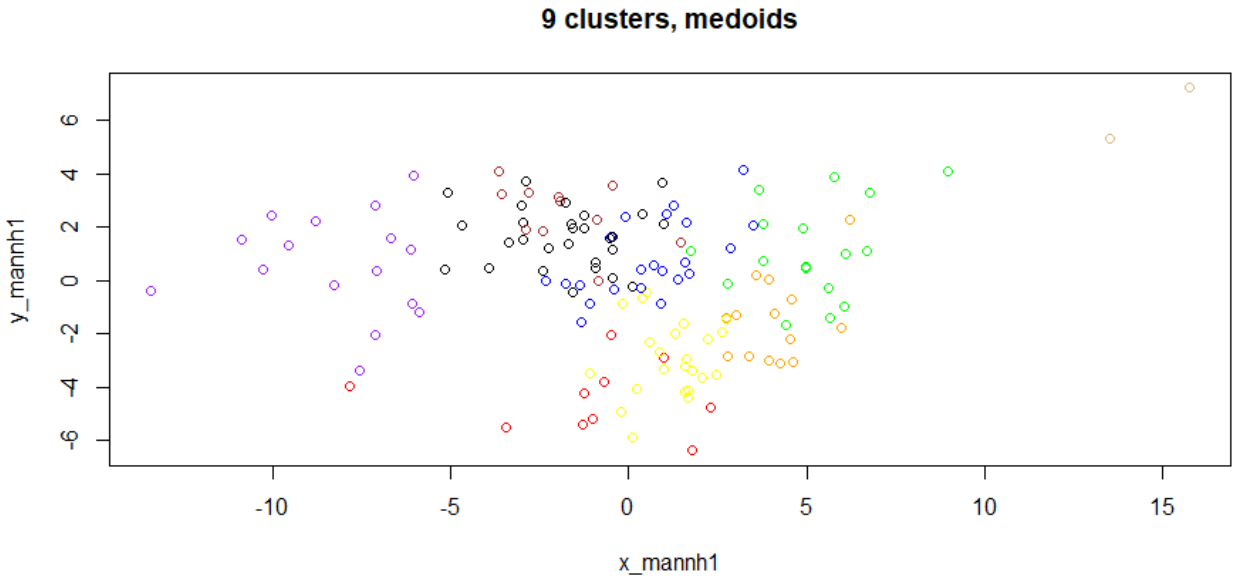
Легко бачити, як на останніх двох діаграмах хаотично розфарбовуються в кластери точки. Як і було припущено в першій роботі, така кількість кластерів

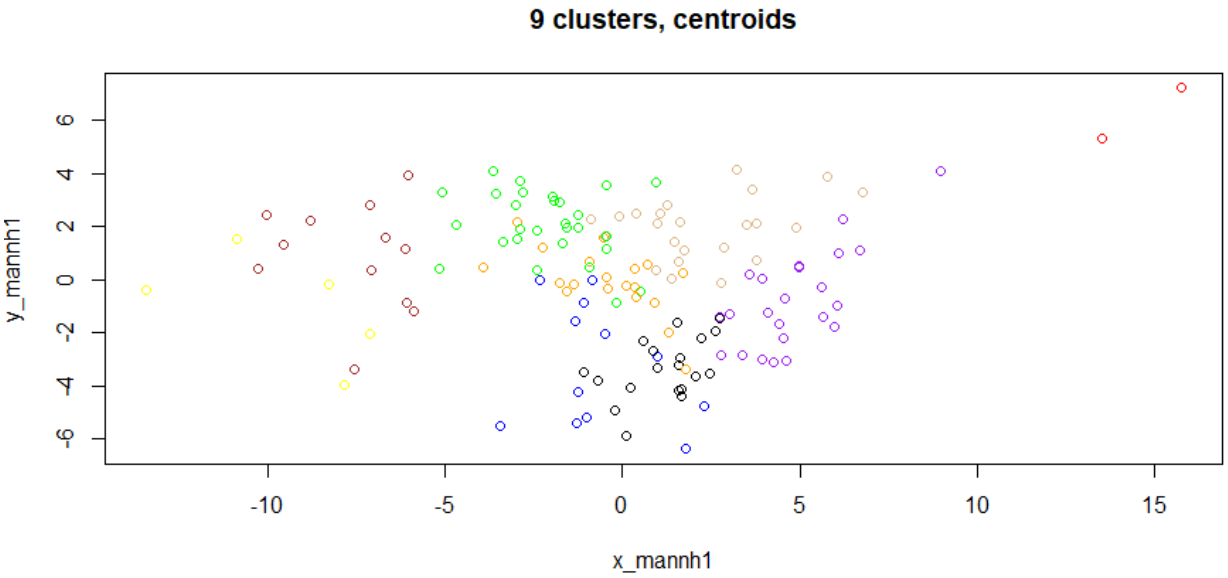
не є доцільною. І хоча у випадку 4-х кластерів розфарбування виглядає найбільш адекватним, проте, на мою думку, воно не є ідеальним. Але серед даних однозначно є.

Для манхаттанської відстані:

```
> # манхаттанська відстань
> x_mannh1 <- fit_mannh1$points[,1]
> y_mannh1 <- fit_mannh1$points[,2]
> # k = 4, medoids
> plot(x_mannh1, y_mannh1, col = col[pam.res41$cluster], cex = 1, main = '4 c
lusters, medoids')
> # k = 9, medoids
> plot(x_mannh1, y_mannh1, col = col[pam.res91$cluster], cex = 1, main = '9 c
lusters, medoids')
> # k = 9, centroids
> plot(x_mannh1, y_mannh1, col = col[km.res91$cluster], cex = 1, main = '9 cl
usters, centroids')
```



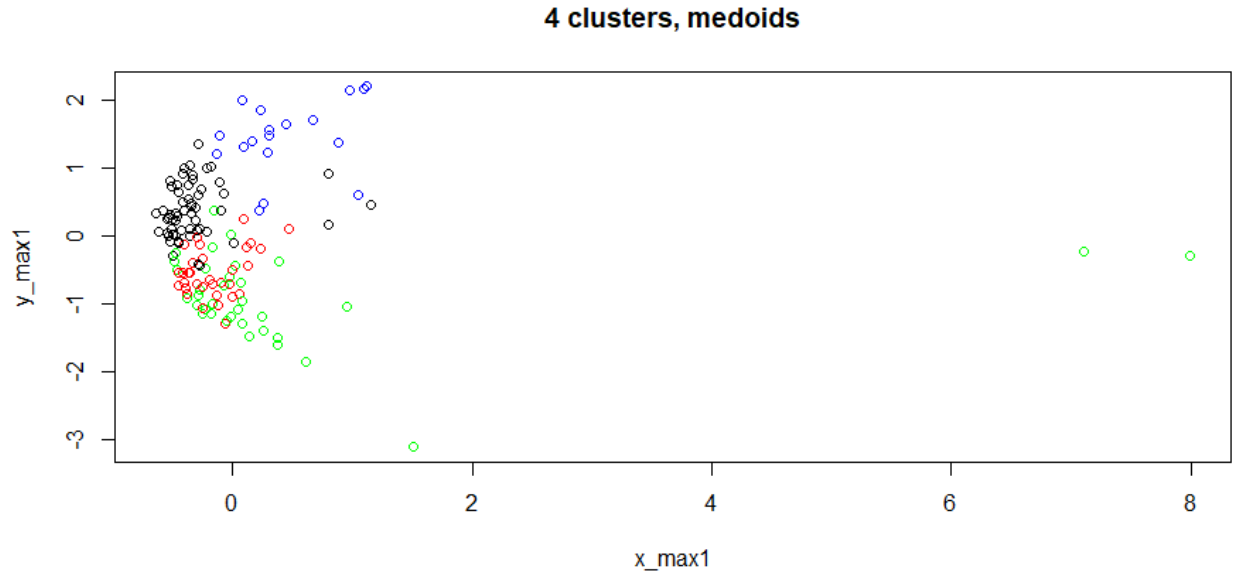


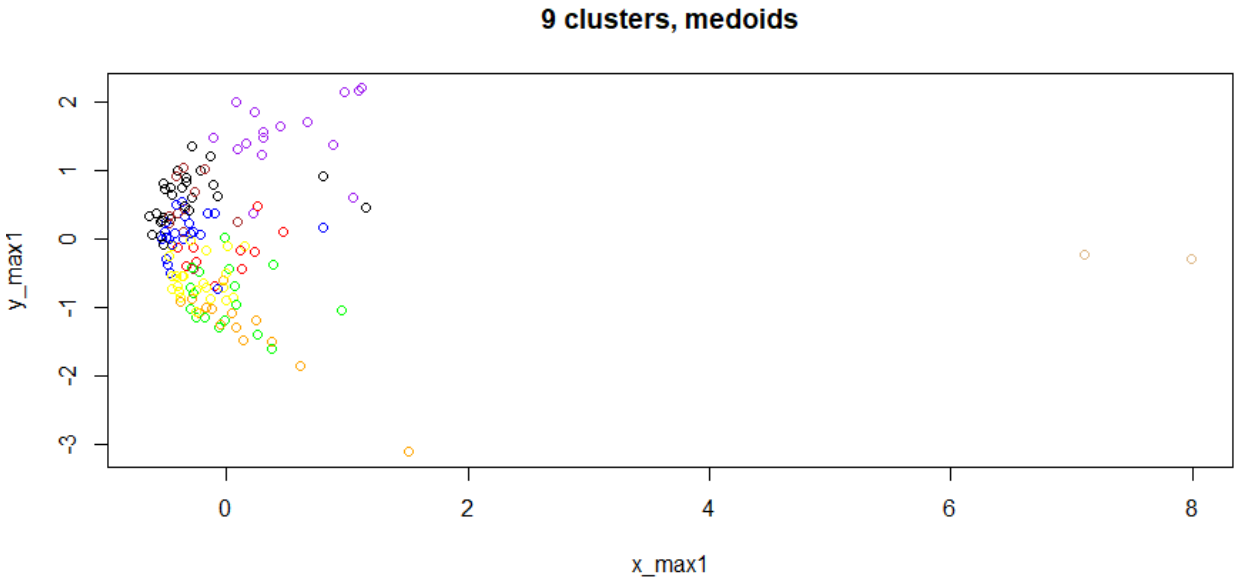


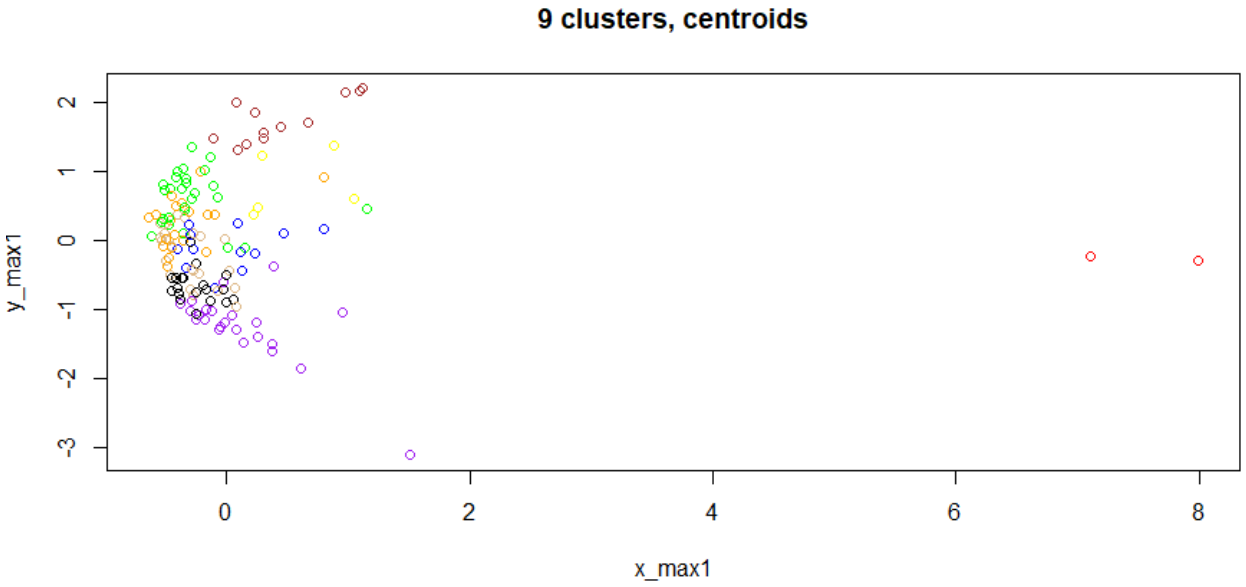
Знову можемо спостерігати незадовільну поведінку у випадку поділу на 9 кластерів, і, ну хоча б трохи прийнятну поведінку у випадку 4.

Спробуємо максимальну відстань:

```
> # максимальна відстань
> x_max1 <- fit_max1$points[,1]
> y_max1 <- fit_max1$points[,2]
> # k = 4, medoids
> plot(x_max1, y_max1, col = col[pam.res41$cluster], cex = 1, main = '4 clust
ers, medoids')
> # k = 9, medoids
> plot(x_max1, y_max1, col = col[pam.res91$cluster], cex = 1, main = '9 clust
ers, medoids')
> # k = 9, centroids
> plot(x_max1, y_max1, col = col[km.res91$cluster], cex = 1, main = '9 cluste
rs, centroids')
```







Враховуючи специфіку наших даних, а саме їх центрування і нормування, використання такої дистанції як максимальна не виявилось зовсім виправданим. Виділити бодай яку-небудь структуру в тих даних дуже важко.

Отже, поглянувши на результати класичного багатовимірного шкалування для практичних даних, я особисто підтвердив би власні здогадки про доцільність розбиття на 4 кластери (методом медоїдів).

2 частина

Тепер розробимо скрипт, що реалізовуватиме канонічно-кореляційний аналіз (CCA) до конфігурації точок, отриманих методом класичного багатовимірного шкалування у просторі великої вимірності (вимірність простору даних першої частини 1 роботи 30, візьмемо, наприклад, 20).

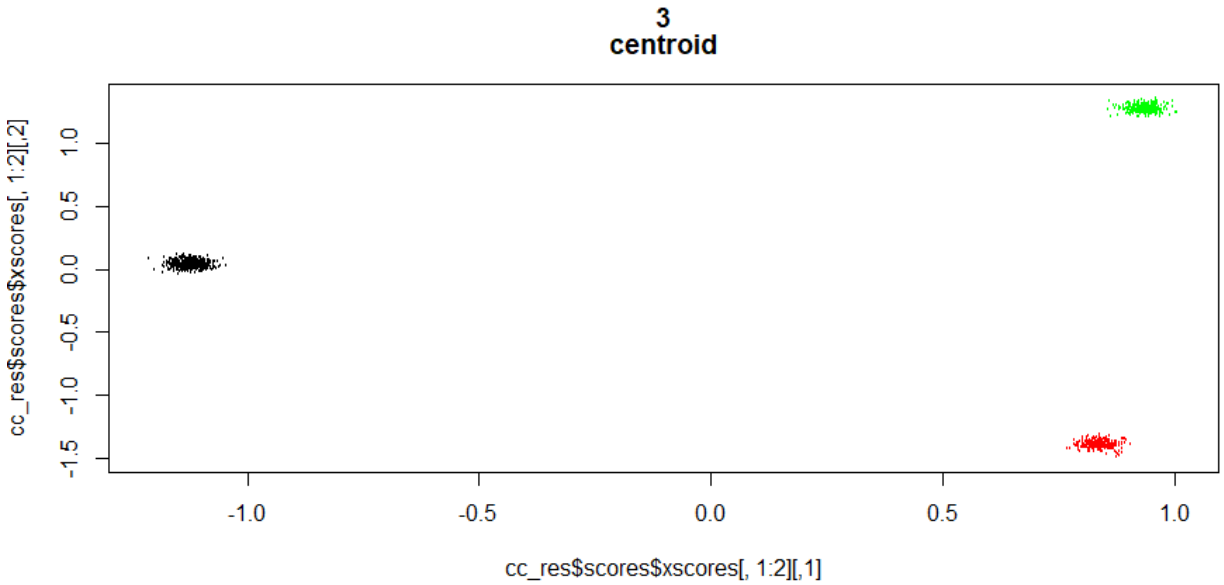
Спершу, реалізуємо відповідну функцію, яка на вхід прийматиме самі дані, кількість кластерів, метод кластеризації, а також варіант відстані, і видаватиме

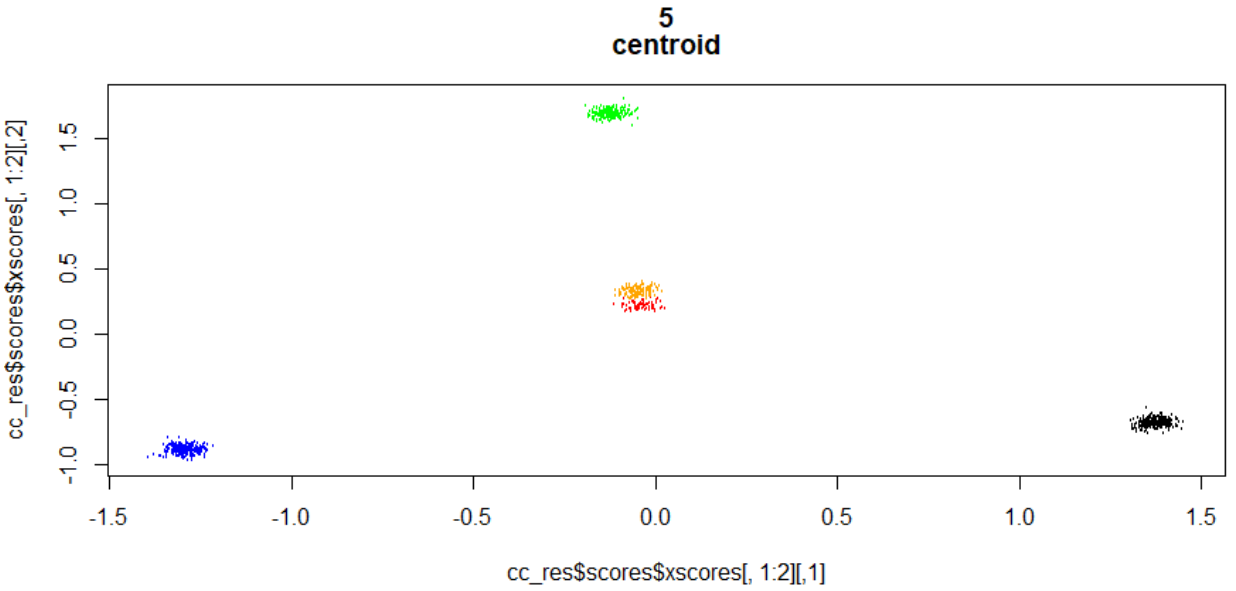
діаграму розсіювання перших двох канонічних компонент із відповідним розфарбуванням на кластери.

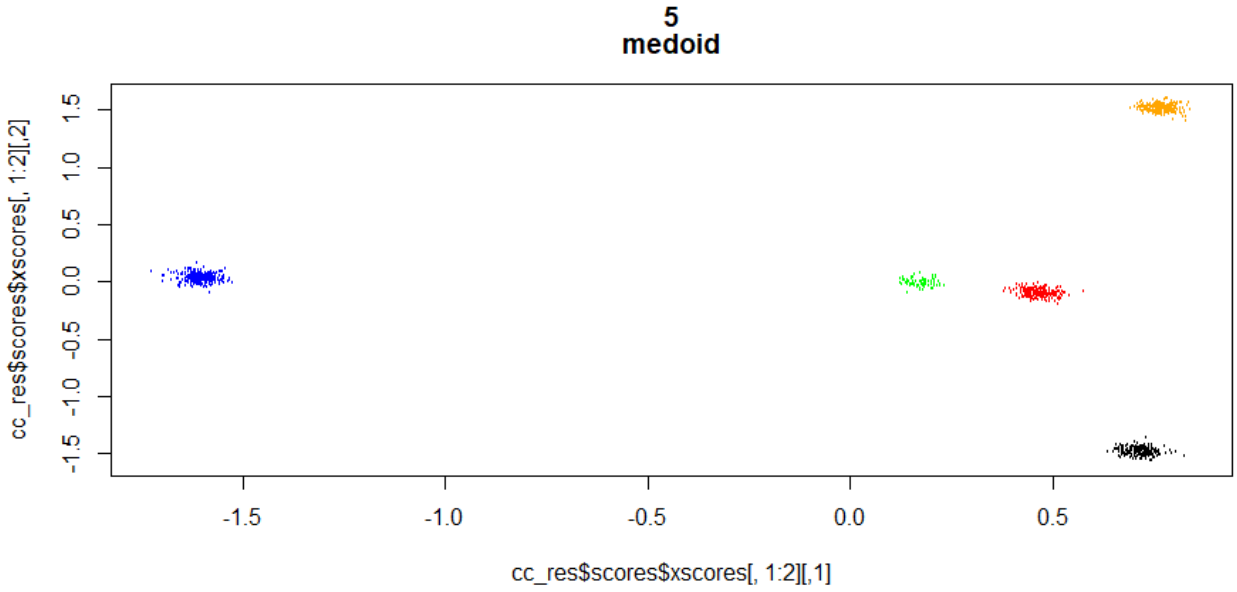
```
> #
> # 2 частина
> #
>
> cca_cmd_plot <- function(data, k, cl_method, dist_function){
+   if(cl_method == 'centroid'){
+     clust <- kmeans(data, k, nstart = 25)
+   } else if(cl_method == 'medoid'){
+     clust <- pam(data, k)
+   }
+   d <- dist(data, method = dist_function)
+   cl <- clust$cluster
+   k <- length(levels(as.factor(cl)))
+   data_cms <- cmdscale(d, k = 20, eig = TRUE)$points
+   n <- nrow(data_cms)
+   C <- matrix(data = as.numeric(rep(cl, k) == rep(1:k, each = n)), ncol = k
, nrow = n)
+   cc_res <- rcc(data_cms,C,0.1,0.1)
+   # діаграма розсіювання перших двох канонічних компонент
+   plot(cc_res$scores$xscores[,1:2], col = col[cl], cex=0.2, main = c(k, cl_
method))
+ }
```

Поглянемо на результати спершу у випадку евклідової відстані.

```
> # евклідова відстань
> cca_cmd_plot(data, 3, 'centroid', 'euclidean')
> cca_cmd_plot(data, 5, 'centroid', 'euclidean')
> cca_cmd_plot(data, 5, 'medoid', 'euclidean')
```



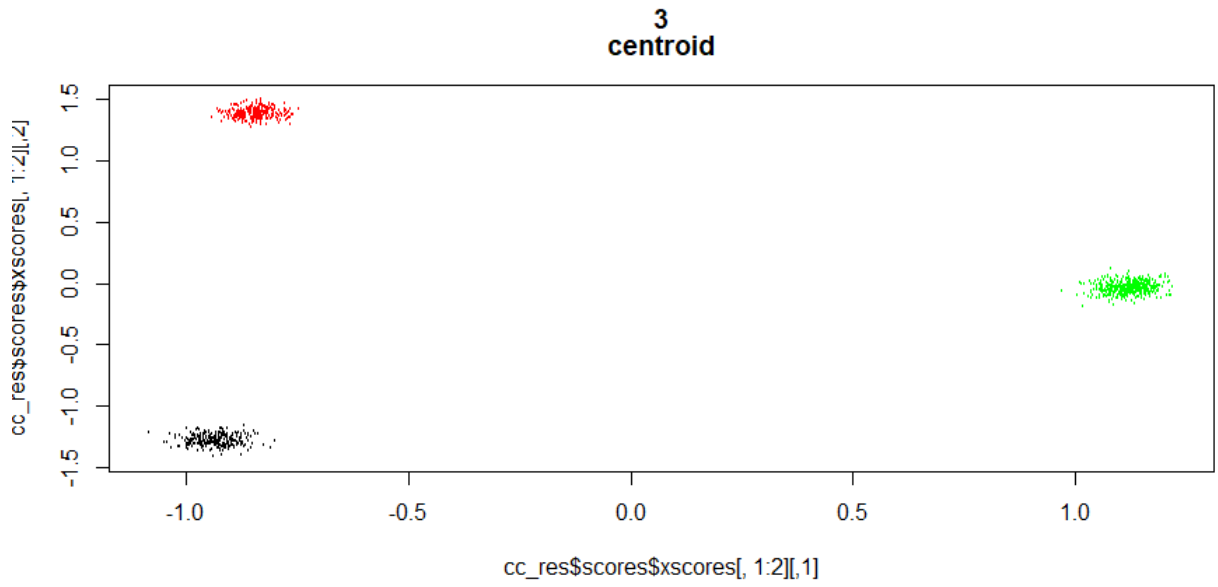


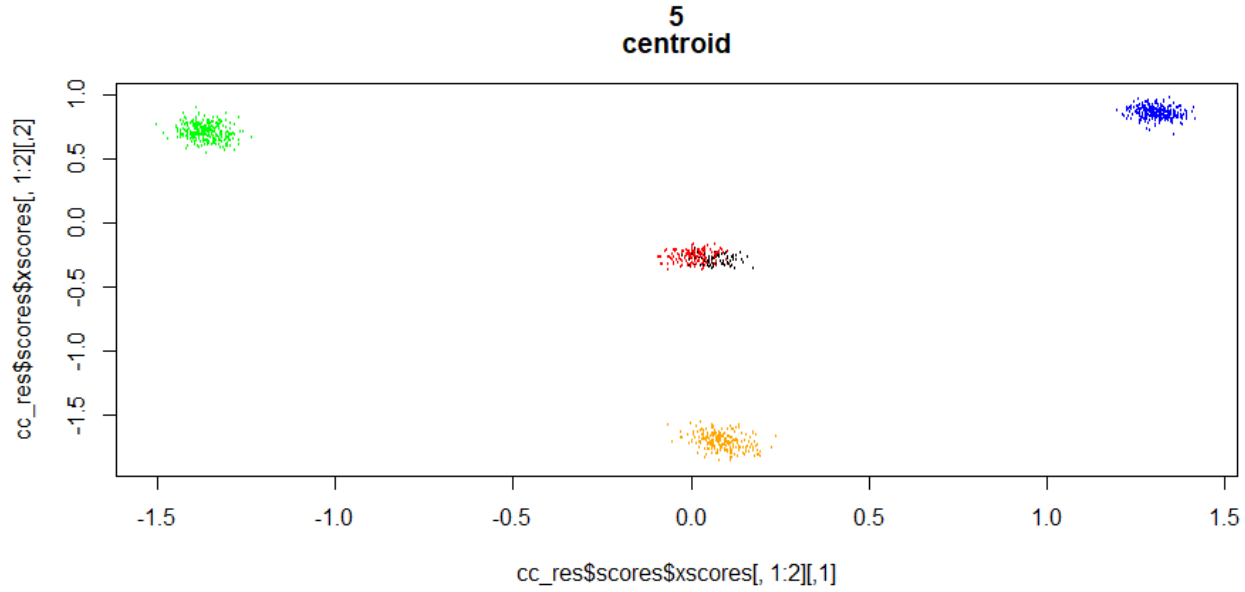


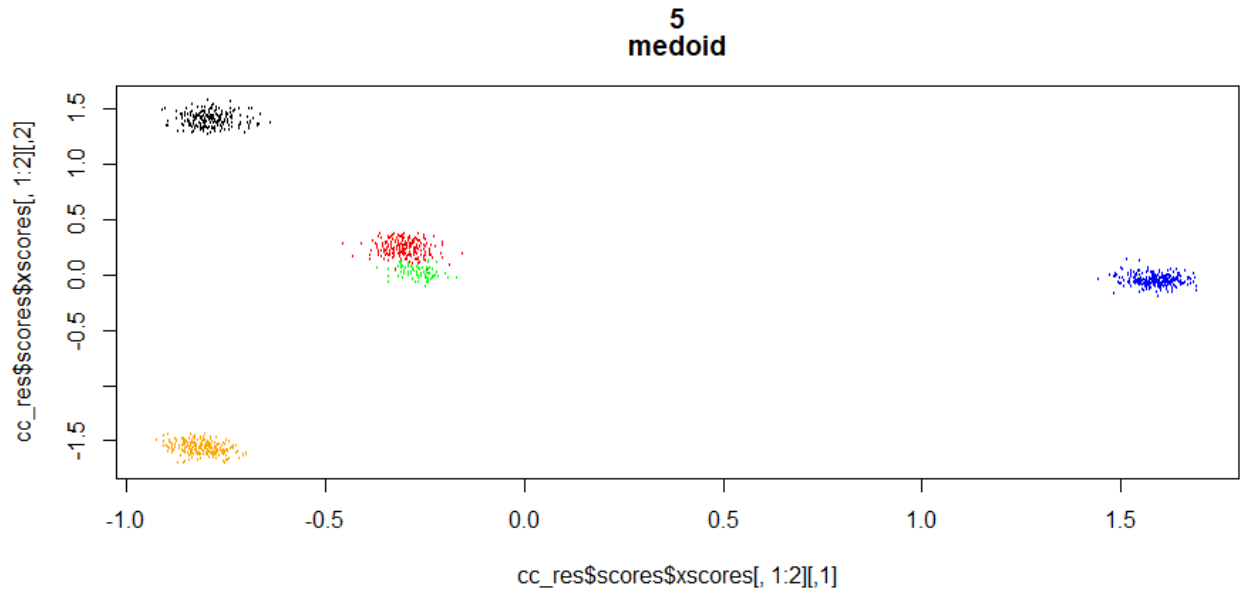
Для евклідової відстані, здається, підтверджується гарний вибір трьох кластерів, і також 5 кластерів для методу медоїдів. Для випадку 5 кластерів методом центроїдів маємо посередині одну двокольорову купку, що не є добре.

Тепер для манхаттанської відстані.

```
> # манхаттанська відстань
> cca_cmd_plot(data, 3, 'centroid', 'manhattan')
> cca_cmd_plot(data, 5, 'centroid', 'manhattan')
> cca_cmd_plot(data, 5, 'medoid', 'manhattan')
```


3
centroid
cc_res$scores$xscores[, 1:2][,2]
cc_res$scores$xscores[, 1:2][,1]

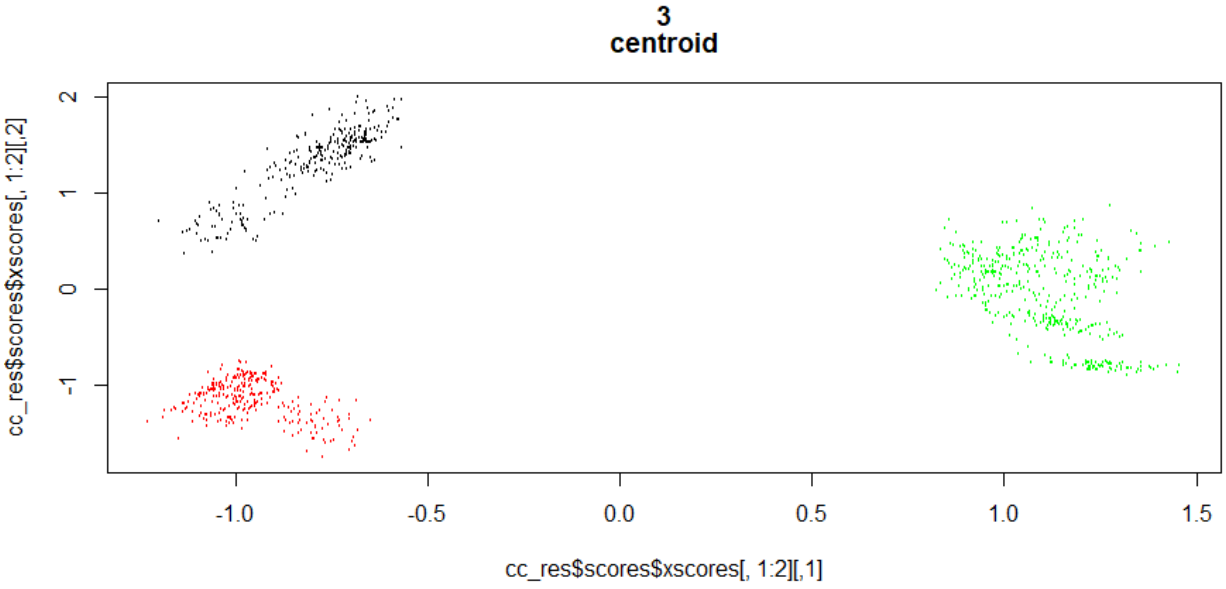

5
centroid
cc_res$scores$xscores[, 1:2][,2]
cc_res$scores$xscores[, 1:2][,1]

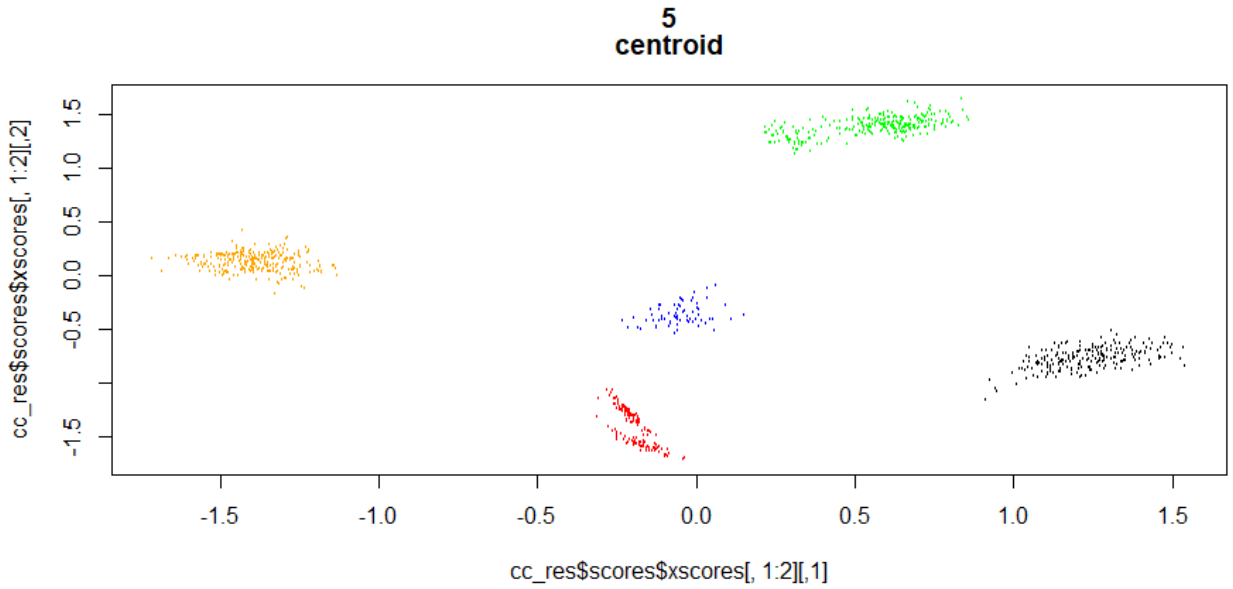

5
medoid
cc_res$scores$xscores[, 1:2][,2]
cc_res$scores$xscores[, 1:2][,1]

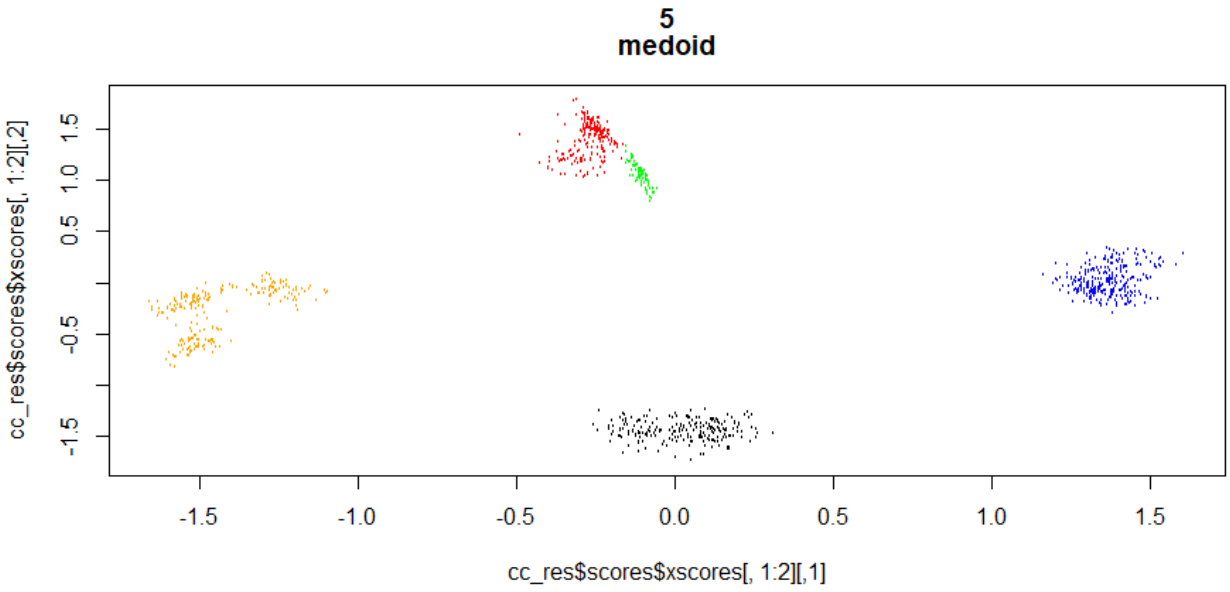
Для манхаттанської ж відстані маємо наступний результат: гарним виявилось розбиття лише на 3 кластери, адже у випадку п’яти (обома методами) маємо одну купку точок, проте двокольорових.

І наостанок застосуємо максимальну відстань.

```
> # максимальна відстань
> cca_cmd_plot(data, 3, 'centroid', 'maximum')
> cca_cmd_plot(data, 5, 'centroid', 'maximum')
> cca_cmd_plot(data, 5, 'medoid', 'maximum')
```







Максимальна відстань єдина показує, що кластеризація на 5 кластерів методом центроїдів є прийнятною, а медоїдів – ні. Але як би там не було, все одно розбиття на 3 кластери виглядає найадекватніше, а отже, це, мабуть, той варіант, на якому варто зупинитись.