Testy poprawności

Michał Wdowski

Wprowadzenie

Tematyką drugiej pracy domowej z przedmiotu *"Przetwarzanie Danych Ustruktyruzowanych"* była analiza skupień *(spectral clustering)*. Jest to problem, który polega na automatycznym umieszczeniu punktów w grupach. Jednym z poleceń wspomnianej pracy domowej była samodzielna implementacja algorytmu spektralnego. Dane, na których zadaniem było testowanie algorytmu to zbiory punktów z przestrzeni $\mathbb{R}^2$ lub $\mathbb{R}^3$. Ze względu na czytelność prezentowania wyników, a także z powodu przyzwolenia ukrytego w znaczeniu niesionym pod słowem *lub*, dane do testowania w niniejszym raporcie będą zbiorami punktów jedynie z przestrzeni $\mathbb{R}^2$.

Omówienie algorytmu

Algorytm zaporponowany w treści zakładał następujące działania:

1. Znalezienie M najbliższych punktów dla każdego punktu
2. Utworzenie macierzy sąsiedztwa grafu i uspójnienie grafu
3. Wyznaczenie k odpowiednich wektorów własnych laplasjanu tego grafu
4. Wykonanie na wektorach algorytmu k średnich.

Algorytm jest wykonywany przez wywołanie funkcji `spectral_clustering(X, k, M)`.

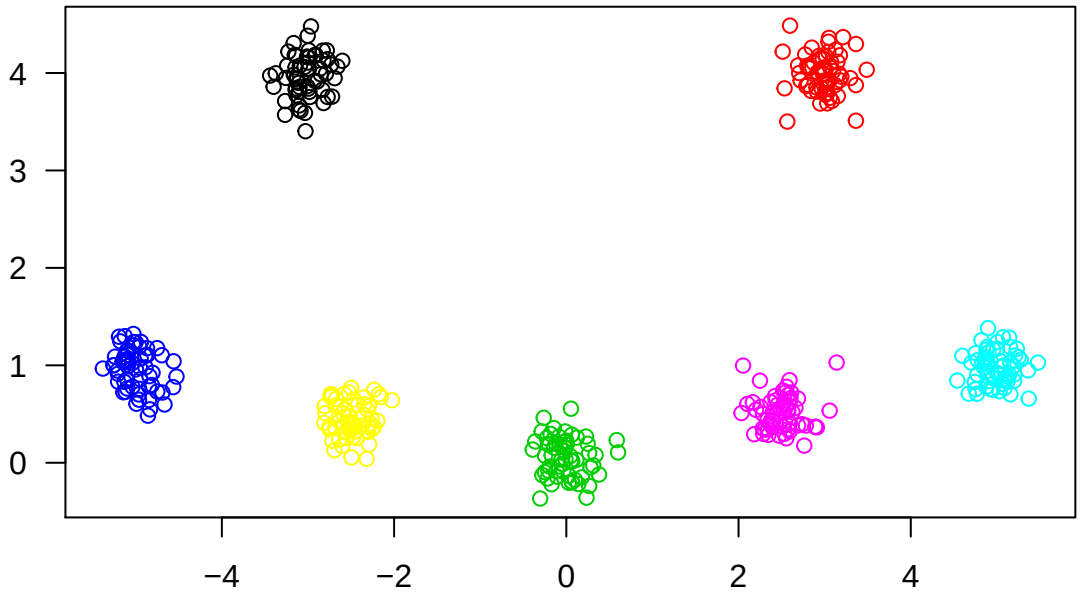
Poprawność działania będzie mierzona za pomocą dwóch indeksów, które porównują dwa zestawy podziałów:

- indeks Fowlkesa–Mallowsa (**FM**)
- skorygowany indeks Randa (**AR**)

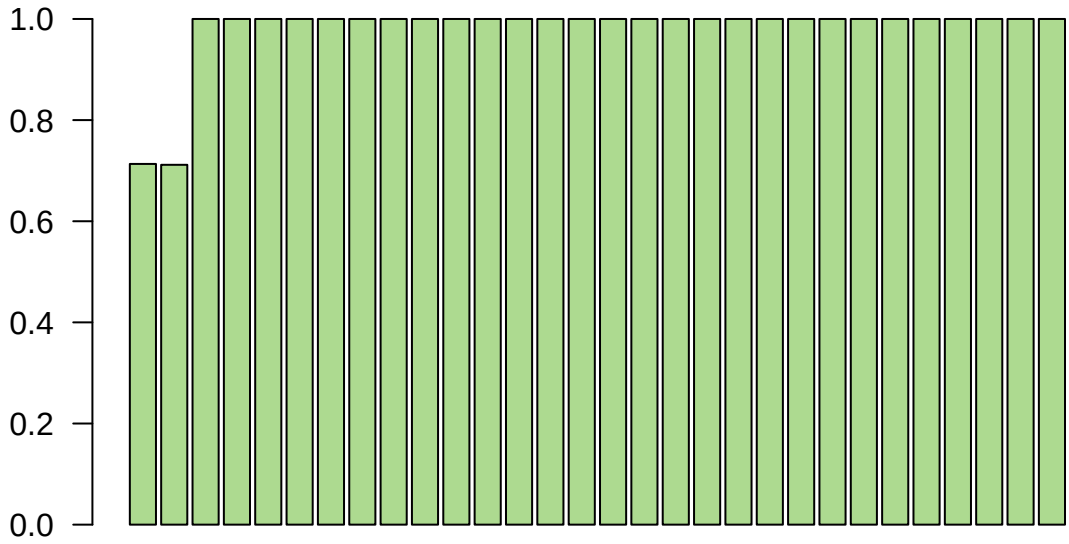
Obydwa indeksy zwracają wartość z zakresu $[0, 1]$, gdzie 1 oznacza idealne pokrycie z docelowym podziałem. Poprawność algorytmu będziemy sprawdzać w zależności od parametru M, który będzie liczbą naturalną z zakresu $[1, 30]$.

Plik `smile.data`

Ten zbiór danych zawiera 420 par współrzędnych punktów w przestrzeni $\mathbb{R}^2$, które układają się w kształt uśmiechu. Poszczególne punkty skupienia zostały pokolorowane według ich przynależności do grup. Każdy z nich zawiera równo 60 punktów, i każdy z nich jest w znaczącej odległości od pozostałych punktów skupienia. Docelowo testowany algorytm powinien stworzyć kolorowanie jak najbardziej podobne do tego:

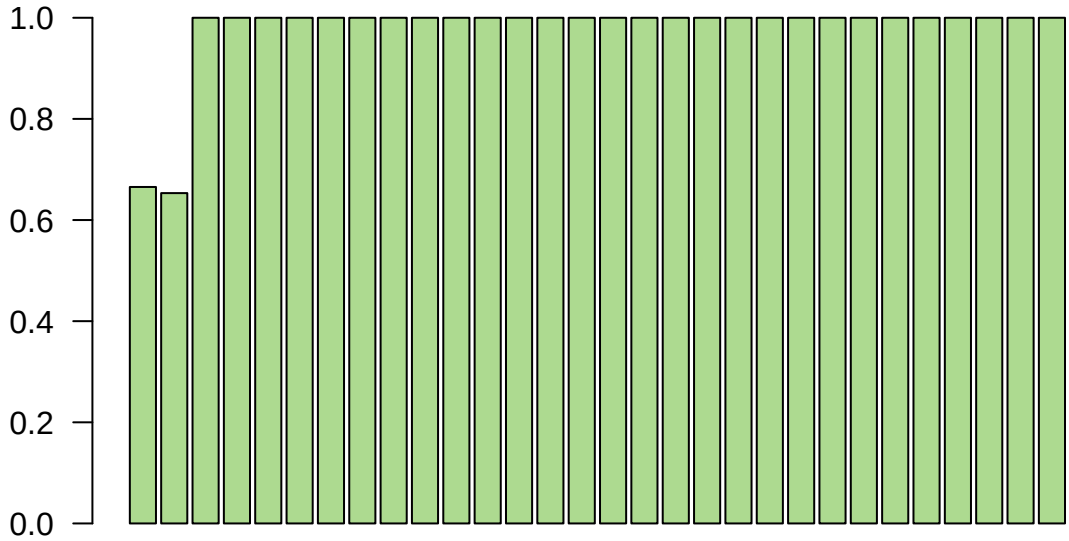


Przyjrzyjmy się indeksom **FM** w zależności od parametru M:

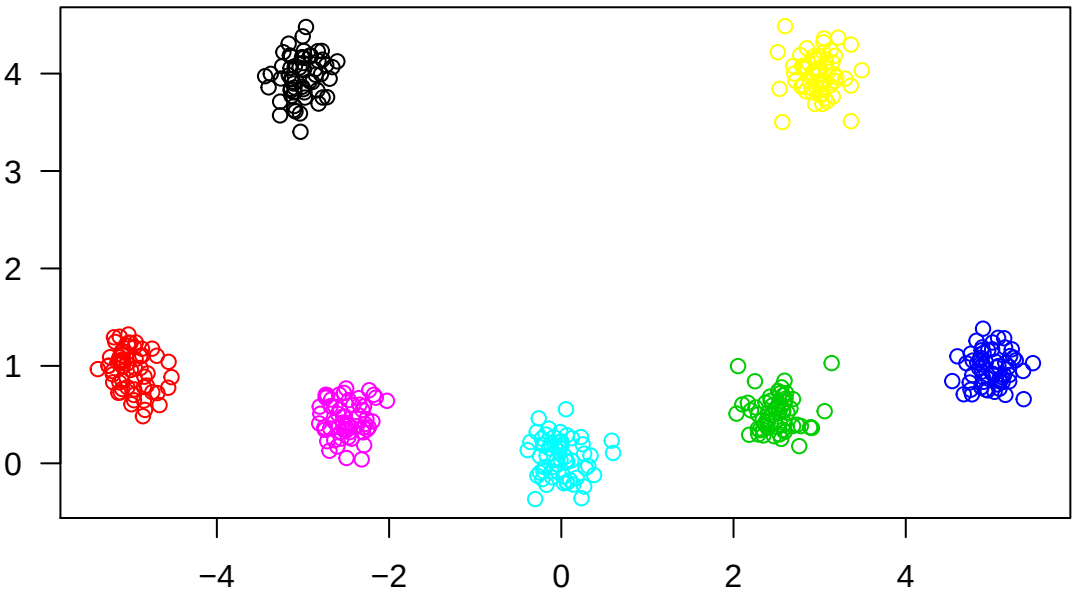


Jak widać, okazuje się, że dla wartości M większych niż 2, z reguły poprawność wykonania algorytmu będzie zadowalająca. Z powodu sposobu działania algorytmu k średnich, który ma w sobie elementy losowe, nie można być nigdy do końca pewnym, czy uda nam się osiągnąć idealne pogrupowanie, nawet dla tak regularnego i oczywistego do podziału zestawu danych jak `smile.data`.

Wartości indeksów **AR** w zależności od M prezentują się niemal identycznie:

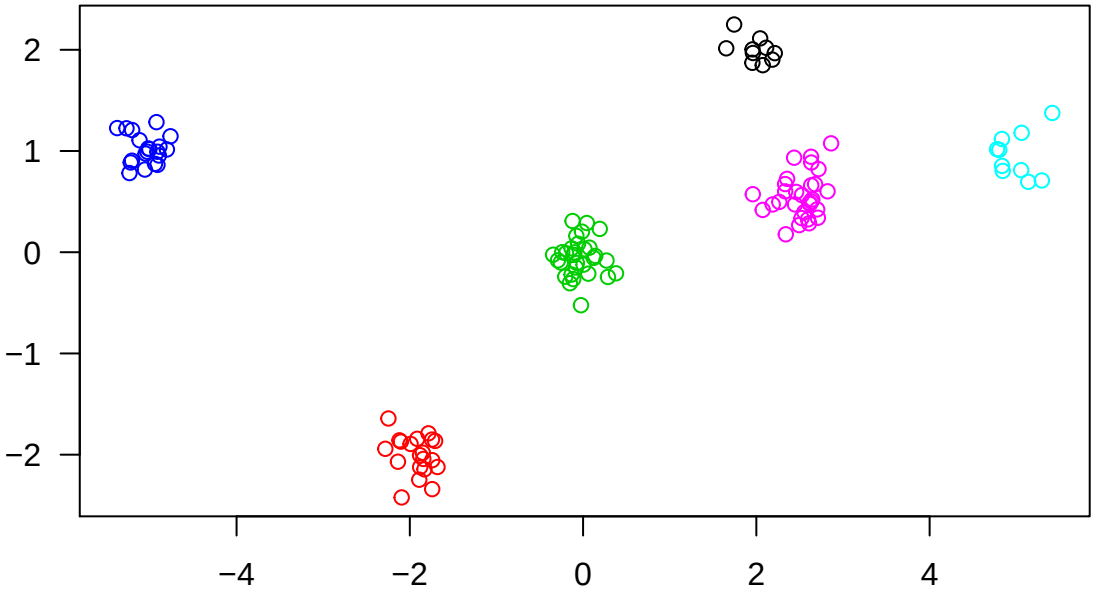


W przypadku tego zestawu danych algorytm sprawdził się bez zastrzeżeń. Kolorowanie pokazane na poniższym rysunku, które reprezentuje podział na grupy dla najlepszego indeksu, pokrywa się z docelowym (z dokładnością do barwy):

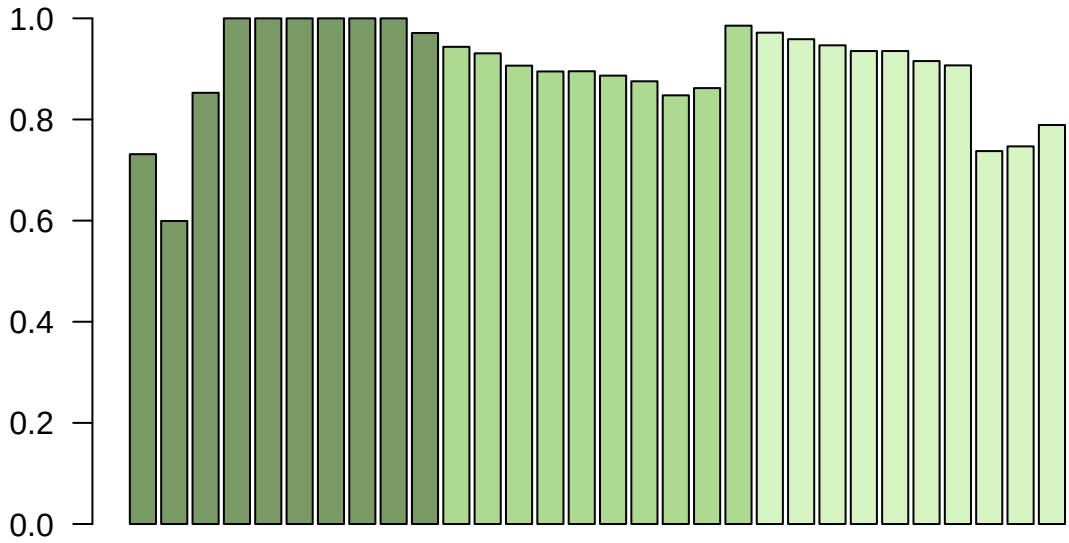


Plik `set.data`

Ten zbiór danych zawiera 120 punktów w przestrzeni $\mathbb{R}^2$, które dzielą się na 6 grup, a każda z nich zawiera 10, 20 lub 30 elementów. Poszczególne grupy są znacząco odległe od siebie. Celem takiej kontrukcji zestawu danych jest sprawdzenie działania algorytmu dla M przekraczającego rozmiar grup, a także zbadanie sposobu grupowania przy nierównej liczbie elementów każdej z grup. Docelowo algorytm powinien wytworzyć taki podział:

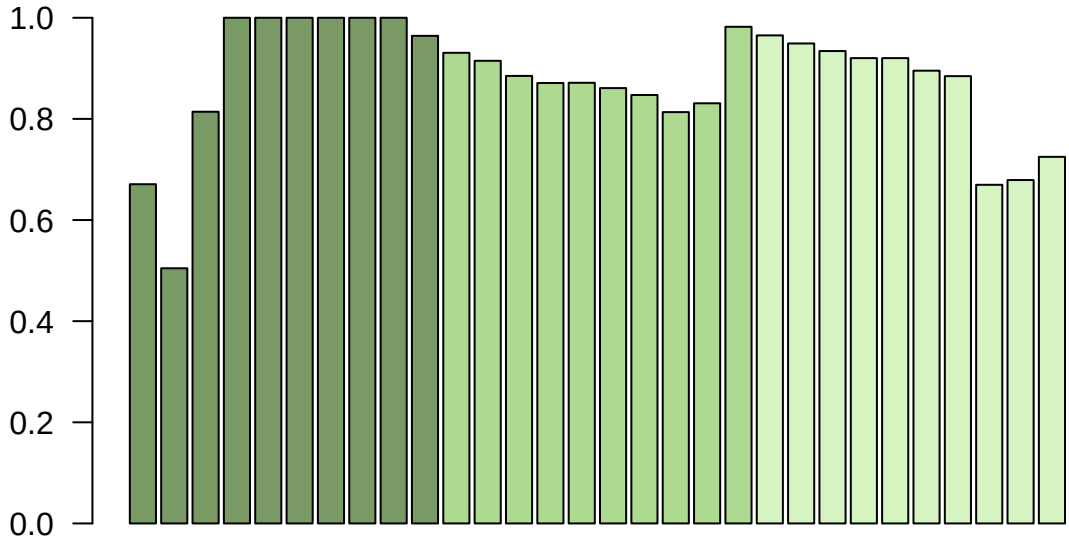


Przyjrzyjmy się indeksom **FM** w zależności od parametru M (odcienie zieleni oznaczają kolejne dziesiątki):

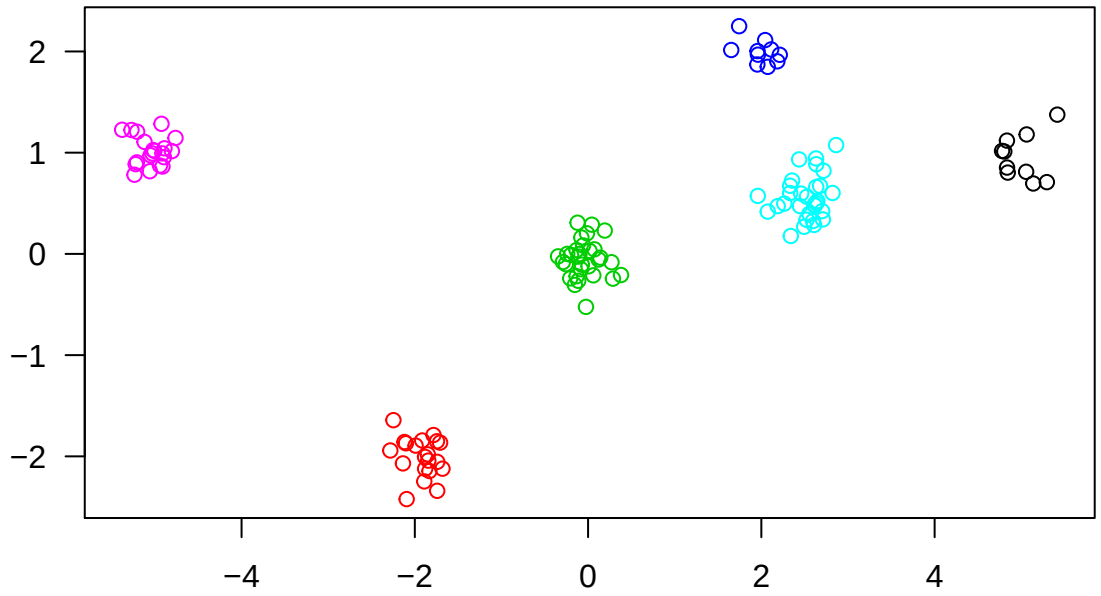


Jak widać, okazuje się, że dla wartości M większych niż 2, z reguły poprawność wykonania algorytmu będzie zadowalająca, dopóki nie osiągnie wartości równej 10, czyli rozmiaru najmniejszej z grup. Co ciekawe, wartość indeksu będzie maleć dla kolejnych M, ale potem dla wartości 20, czyli rozmiaru dwóch grup ze zbioru `set.data`, indeks osiąga maksymalną wartość. Dla następnych M poprawność podziału delikatnie spada, by przy wartości 28 drastycznie spaść. Potem indeksy zaczynają powoli rosnąć.

Wartości indeksów **AR** w zależności od M prezentują się niemal identycznie:

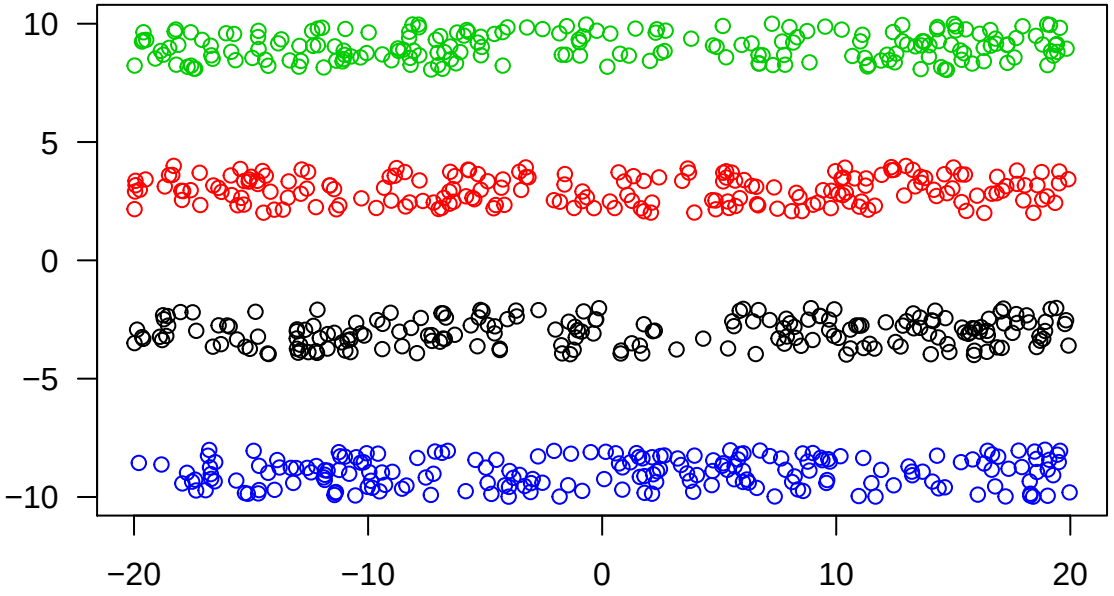


W przypadku tego zestawu danych algorytm sprawdził się dobrze dla M mniejszego od rozmiaru najmniejszej z grup i większego niż 3. Kolorowanie pokazane na poniższym rysunku, które reprezentuje podział na grupy dla najlepszego indeksu, pokrywa się z docelowym (z dokładnością do barwy):

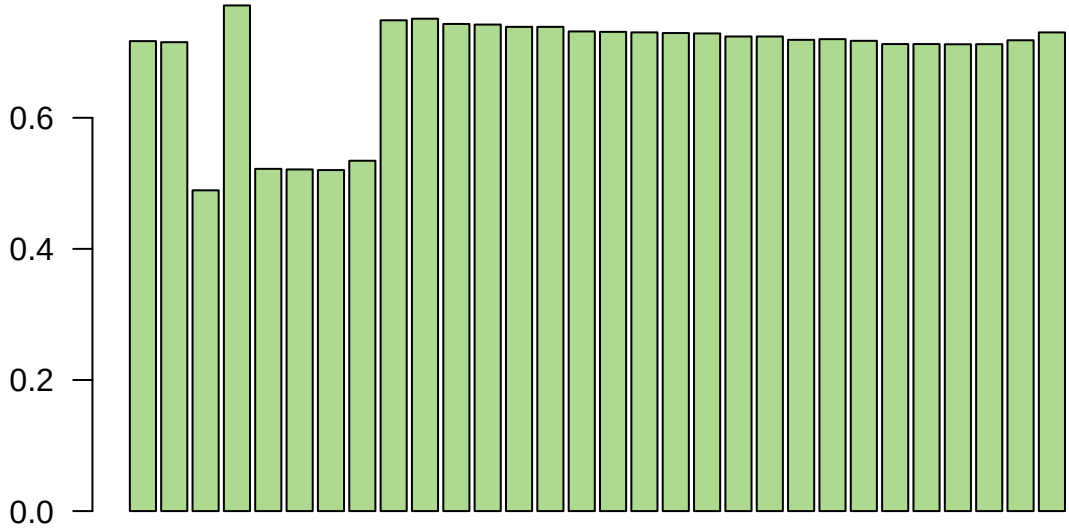


Plik `strips.data`

Ten zbiór zawiera 4 “paski”, każdy po 200 punktów. Są one tak skonstruowane, żeby były blisko siebie. Ma to sprawdzić, czy grupowanie jest tak naprawdę zbiorem punktów w określonej odległości od pewnych środków, czy może podział działa w inny sposób. Docelowe grupowanie ma wyglądać tak:

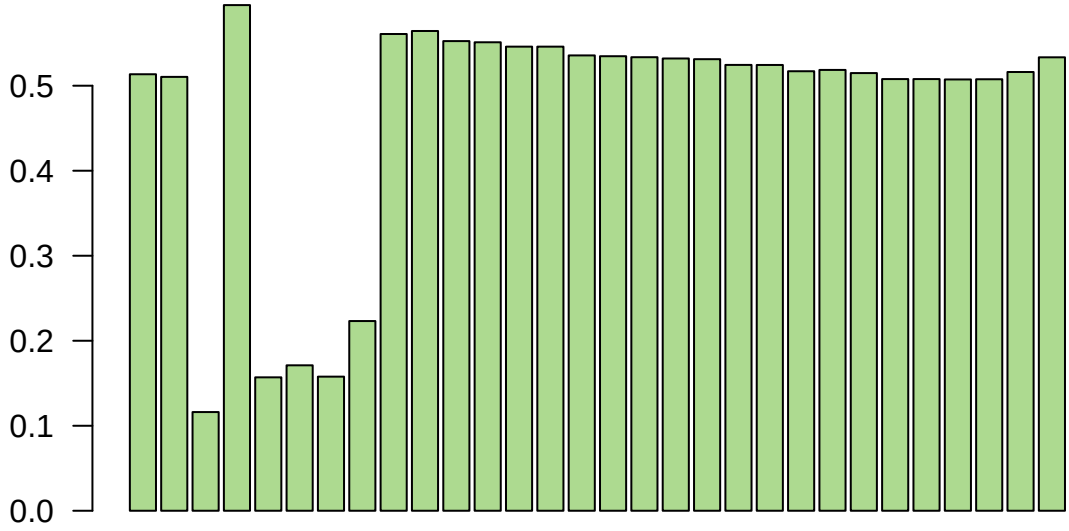


Przyjrzyjmy się indeksom **FM** w zależności od parametru M:



Jak widać, okazuje się, że dla wartości M większych niż 9, z reguły poprawność wykonania algorytmu będzie zadowalająca. Z powodu sposobu działania algorytmu k średnich, który ma w sobie elementy losowe, nie można być nigdy do końca pewnym, czy uda nam się osiągnąć idealne pogrupowanie, widać to po dość chaotycznym zachowaniu algorytmu dla mniejszych M.

Wartości indeksów **AR** w zależności od M prezentują się niemal identycznie:



Nawet przypadku tego zestawu danych algorytm sprawdził się bez zastrzeżeń dla odpowiednio dużych M. Kolorowanie pokazane na poniższym rysunku, które reprezentuje podział na grupy dla najlepszego indeksu, pokrywa się z docelowym (z dokładnością do barwy):