**Задача №2** Автоматическое определение количества кластеров

Необходимо было реализовать метод kmeans с автоматическим определением количества кластеров.

Фреймворк Spark обладает необходимым функционалом, но было решено реализовать все методы самостоятельно ввиду интереса к самим методам. Код написан на Java, графики строились в Matlab.

Я использовал методы kmeans для кластеризации при известном числе кластеров, а так же методы Elbow method (метод локтей) и Silhouette method (метод силуэтов) для определения оптимального (относительно) числа кластеров. Тестировались системы из 500, 1000, 10000 точек.

Сложность метода локтей (включая сам kmeans): O(k\*n\*d\*l)

Сложность метода силуэтов (включая сам kmeans): O(k\*n\*d\*(n + l)), где

k – максимальное количество кластеров (признаков) (подается на вход),

n – количество точек

d = 2 – размерность пространства (у нас точки на плоскости)

l – количество итераций, необходимых для сходимости метода kmeans (верхняя граница определяется числом точек)

k перебиралось в интервале от 1 до 30 – предполагается, что заранее известно максимальное количество классов объектов, на которые они могут быть разбиты.

То может быть совокупность интересов пользователей – в данном случае пользователей нельзя отсортировать четким образом и использование кластеризации оправдано.

**Результаты:**

500 точек, визуально 4-6 кластеров:

- время работы всех методов – менее секунды

- максимальное количество итерации по определению центров - 15

1000 точек, визуально 9 кластеров:

- время работы всех методов – 3,5 секунды

- количество итерации по определению центров не превосходит 40

10000 точек, визуально 9 кластеров:

- время работы метода локтей – 12 секунд

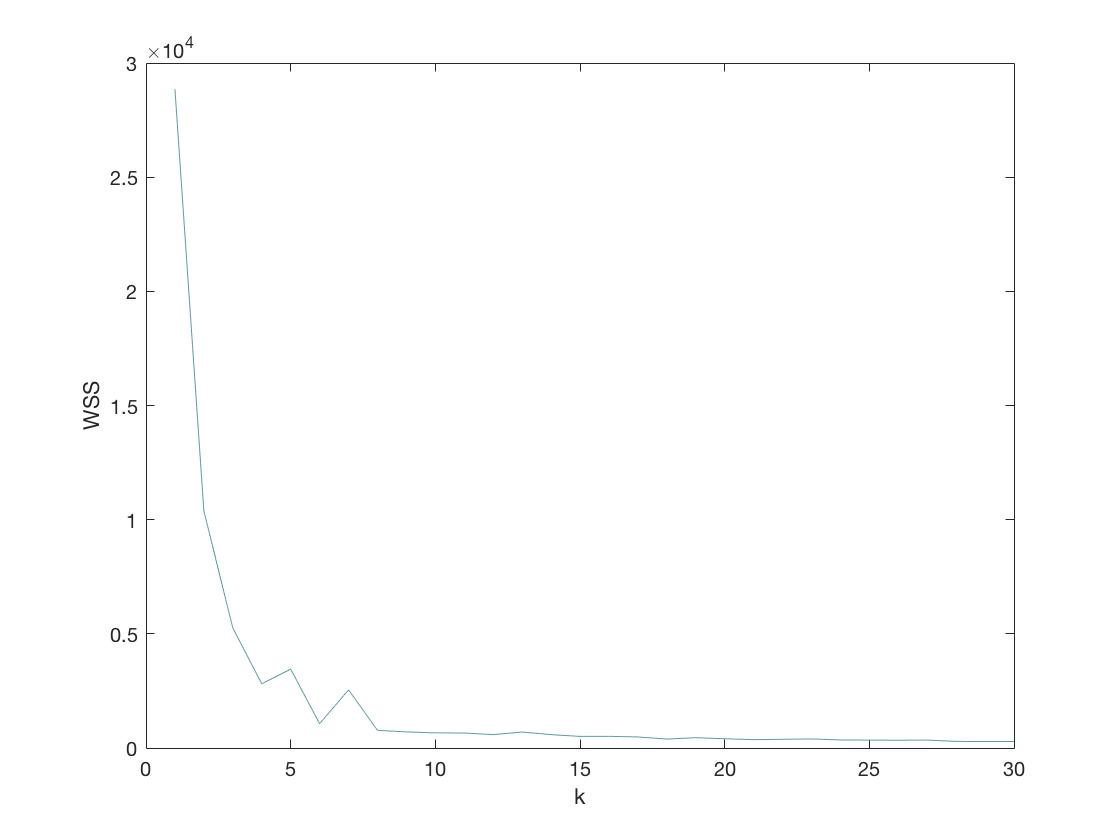
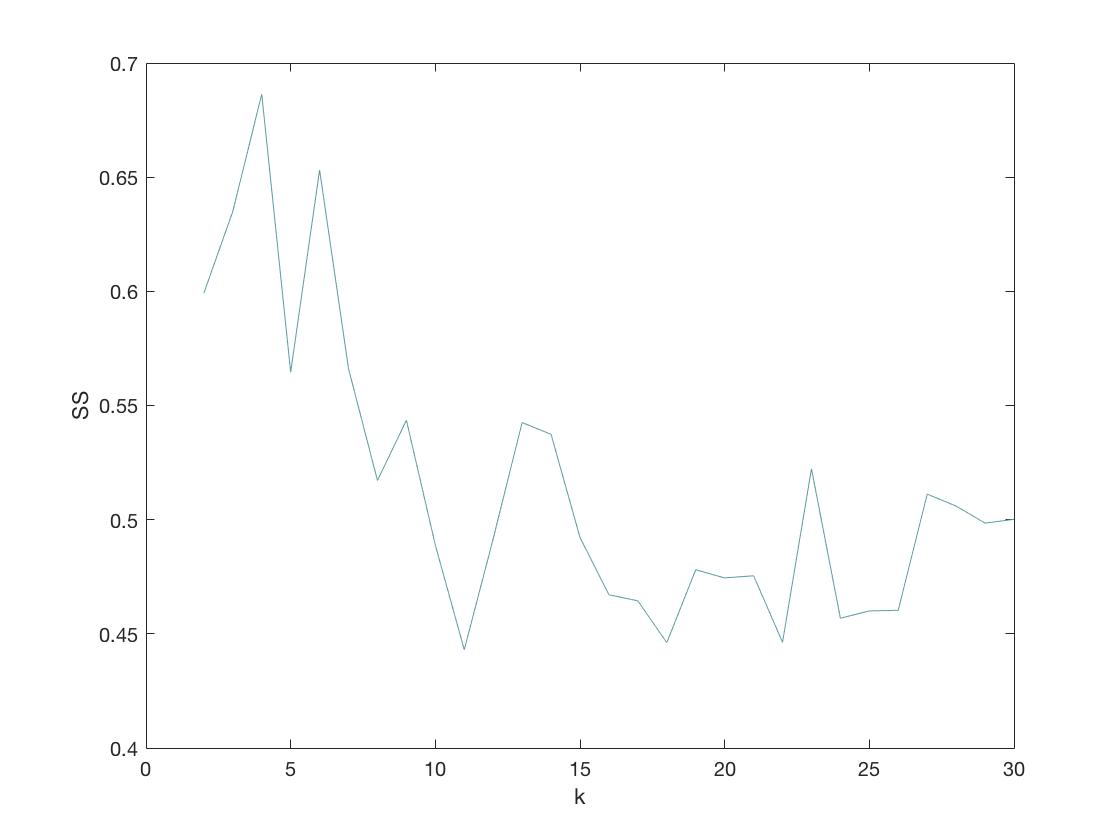
- время работы метода силуэтов – 200 секунд

- количество итерации по определению центров не превосходит 80

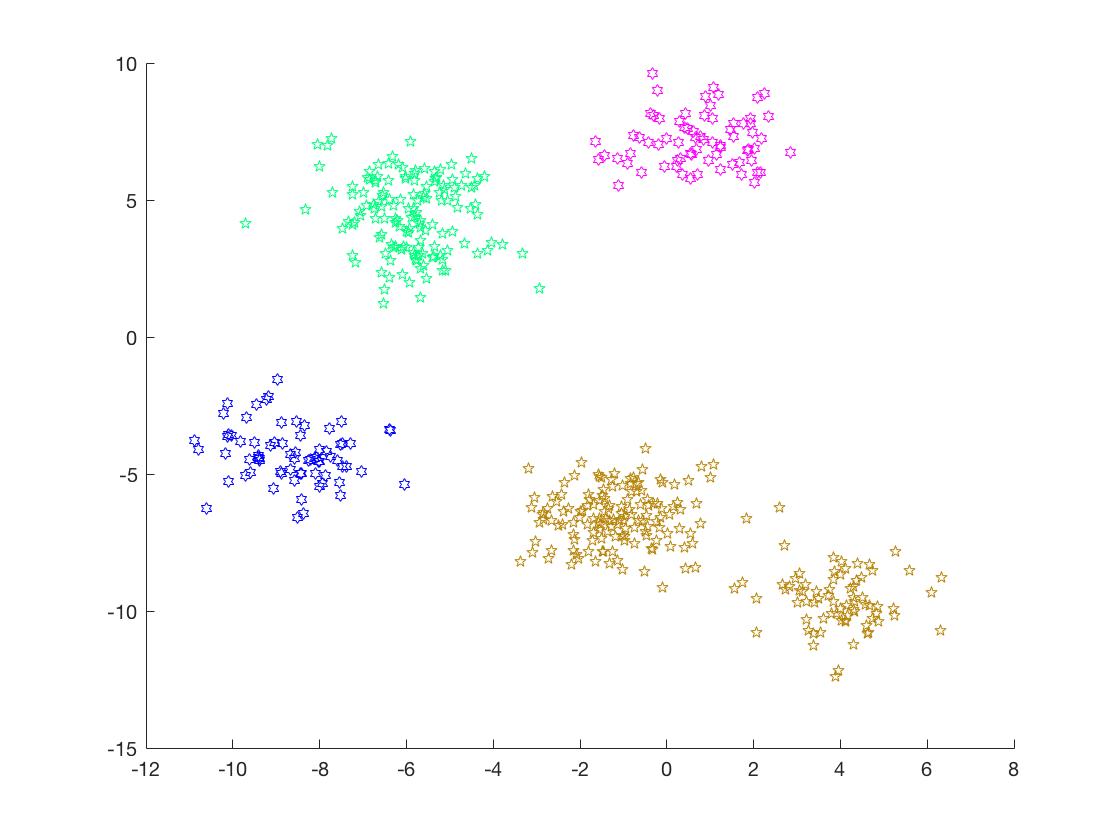
Стоит отметить, что метод локтей визуальный – то есть оптимальное количество кластеров определяется из графика; когда дальнейшее увеличение количества кластеров вносит малый вклад в изменение суммы квадратов отклонений точек от центров их кластеров, k (ось абсцисс) на графике зависимости (within-cluster sum of square от к-ва кластеров k) минует так называемый «локоть».

Метод силуэтов, в свою очередь, возвращает оптимальное k; в методе для каждой точки вычисляется коэффициент, зависящий от “similarity” (сходство) и “dissimilarity” (несходство) точки и определенном ей кластера. Разумеется, нельзя утверждать, что полученное k есть ответ – ответа на задачу кластеризации нет, но он может нас устроить или не устроить. Поэтому мне понравился графический (метод локтей, а так же силуэтов) подход именно к этой задаче, а не всей задачи кластеризации в целом.

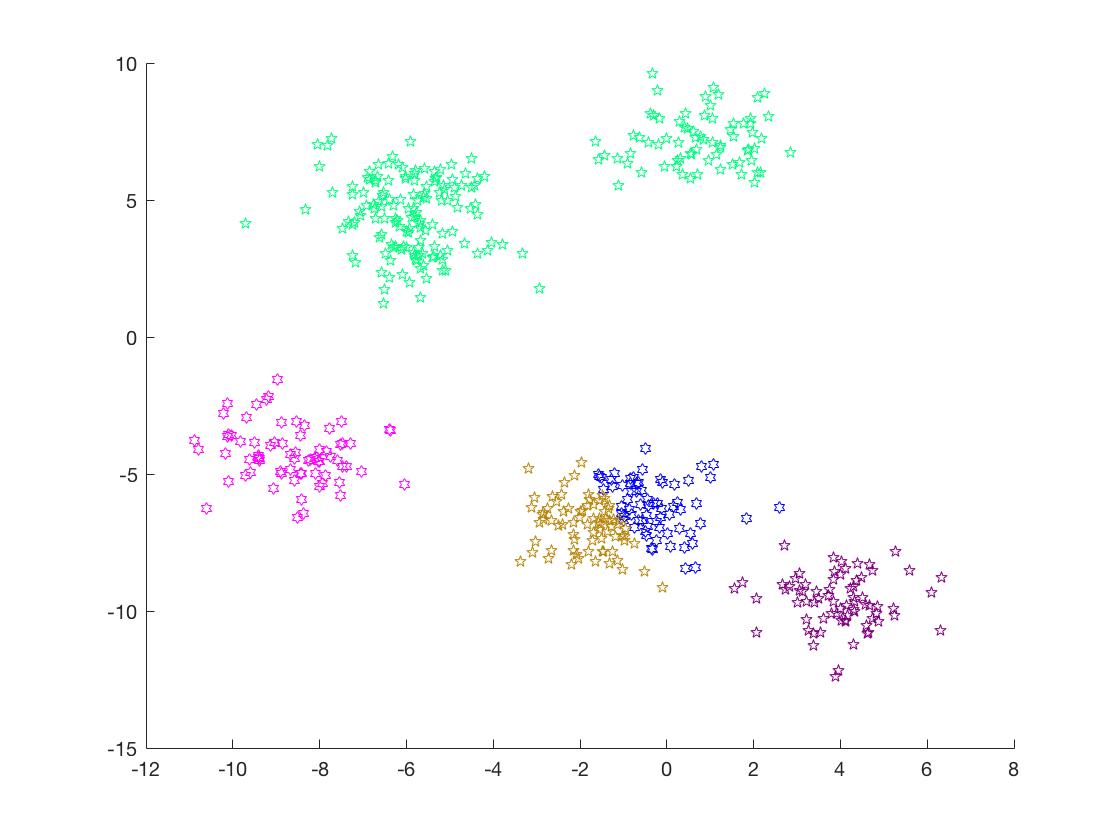
В выходном файле clusterID.txt второй столбец соответствует координате х центра кластера, третий- координате у, а первый- идентификатору кластера. В выходном файле objectID.txt первый столбец соответствует идентификатору точки, второй - идентификатору кластера.

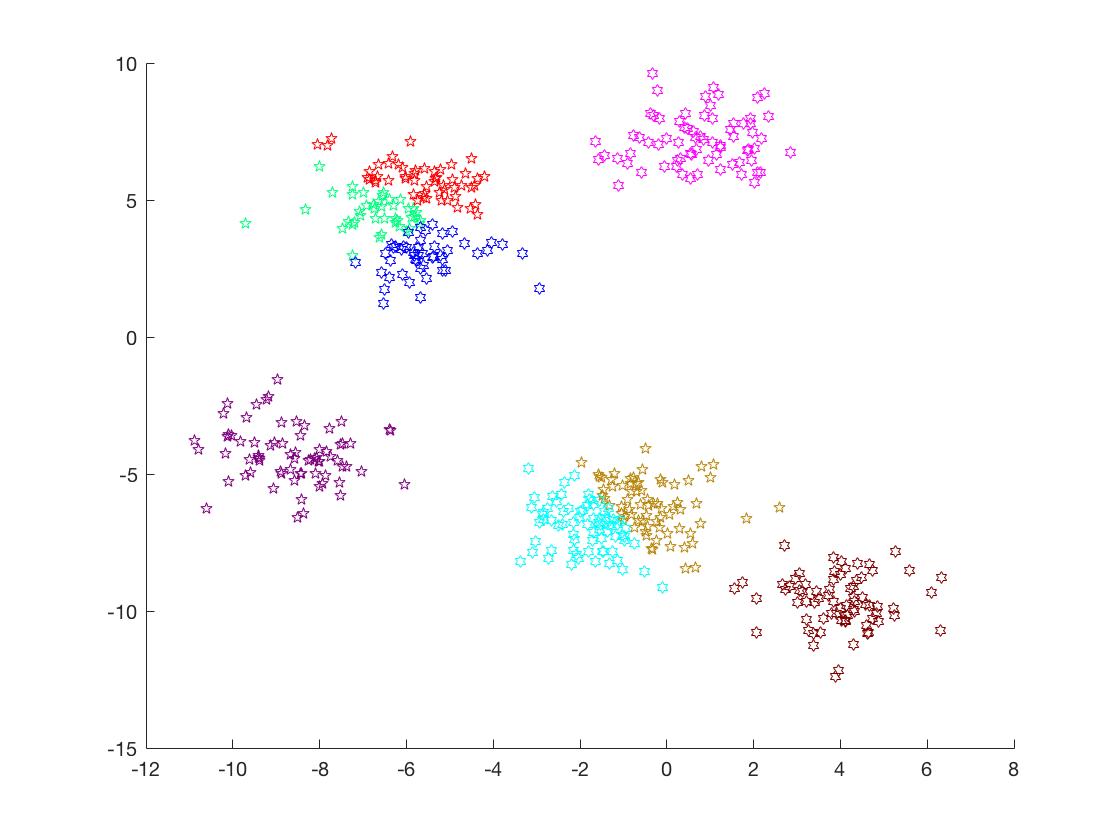


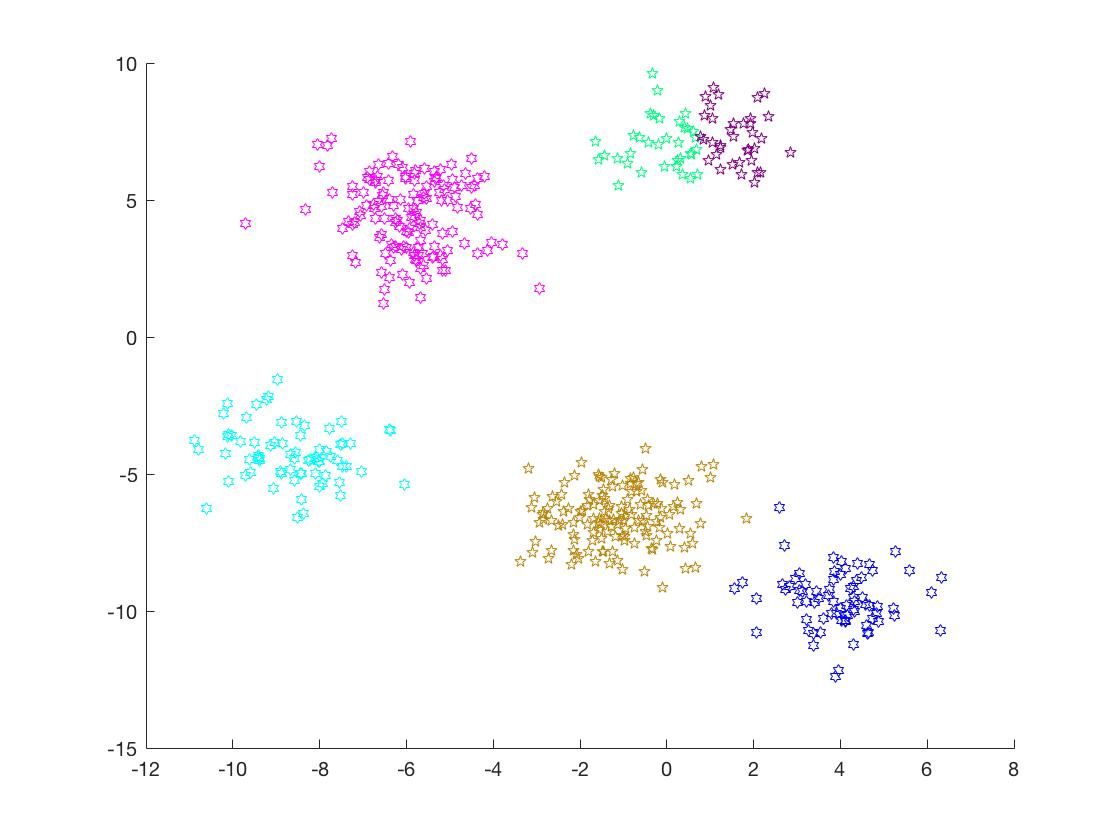
500 точек, графики WSS(k) и SS(k) соответственно



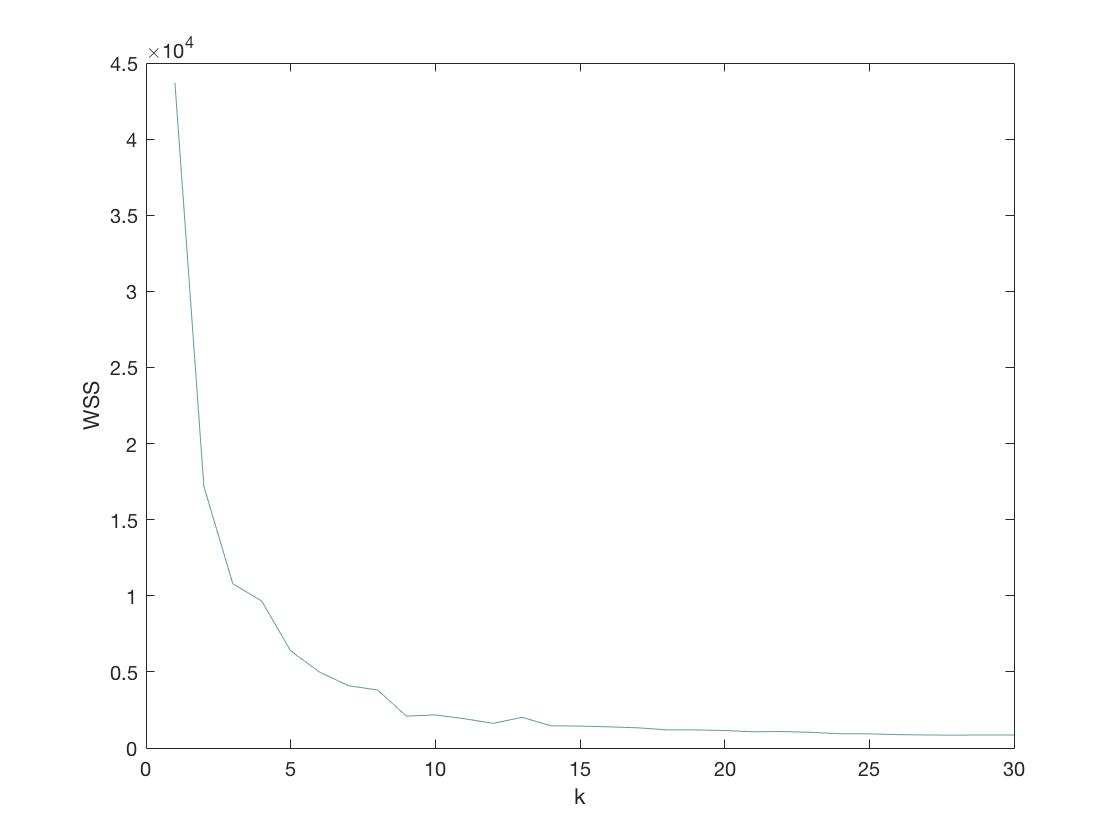
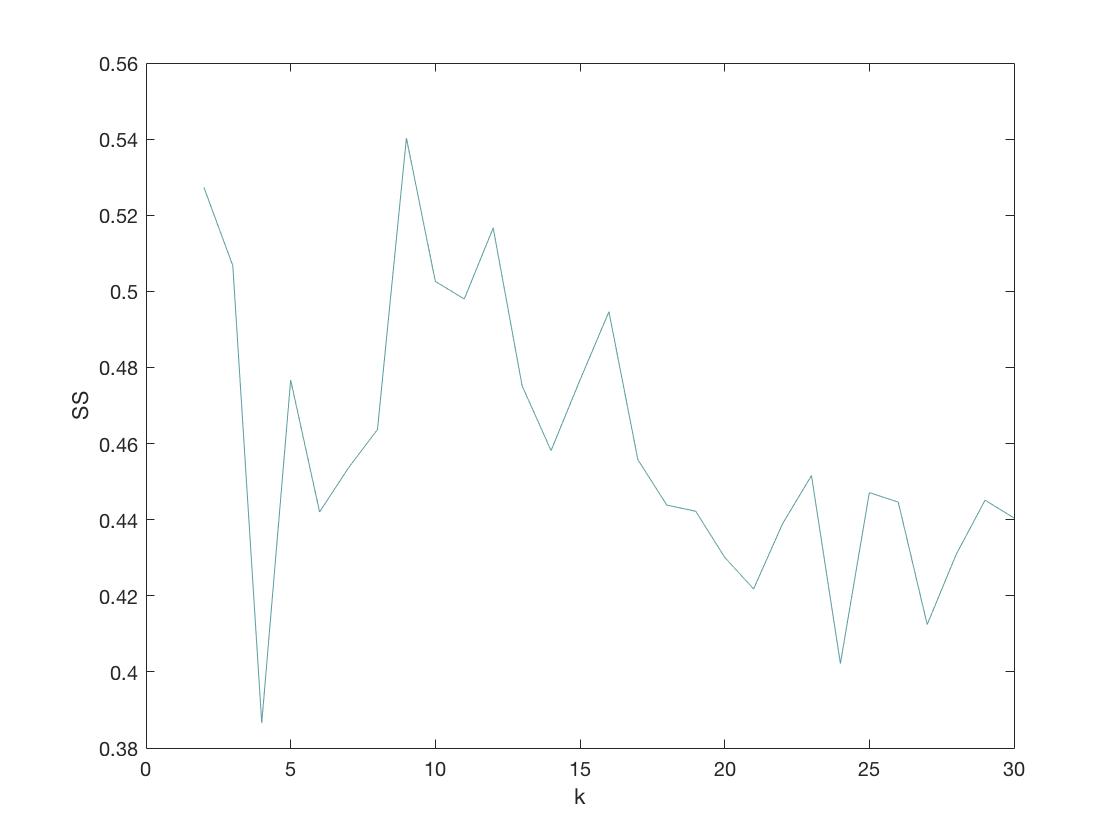
500 точек, k = 6 и k = 4 соответственно



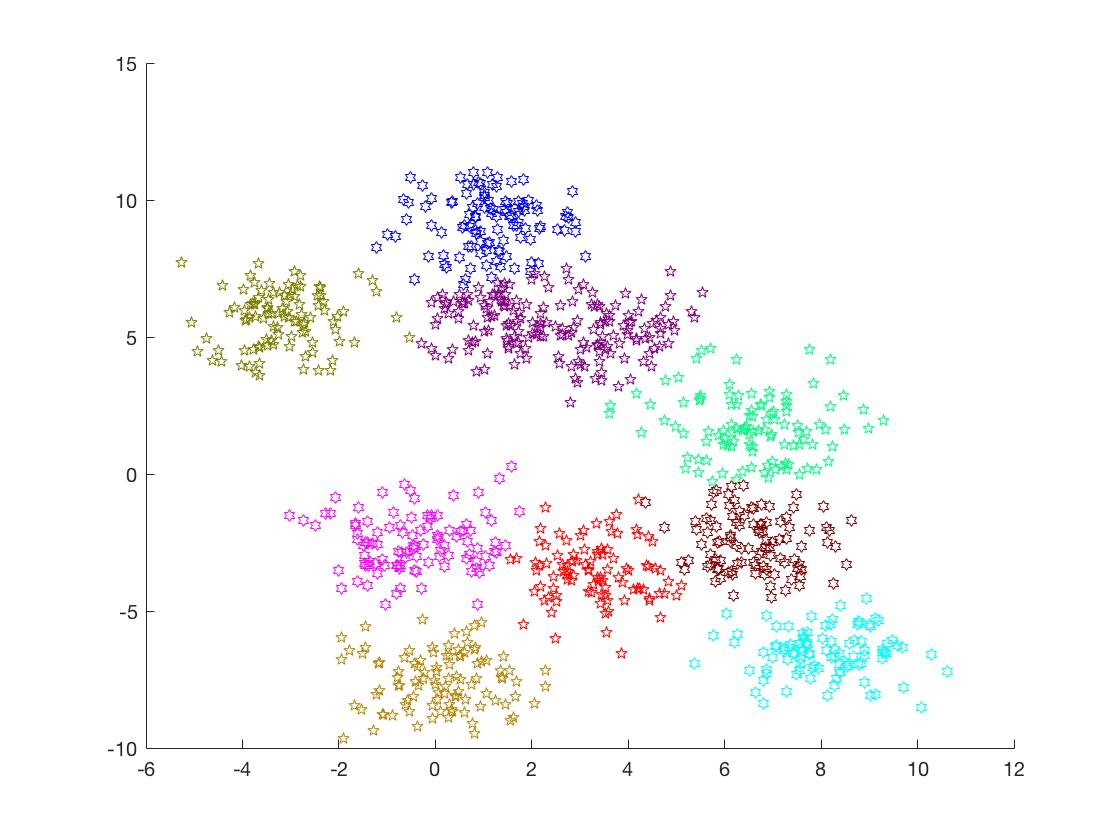
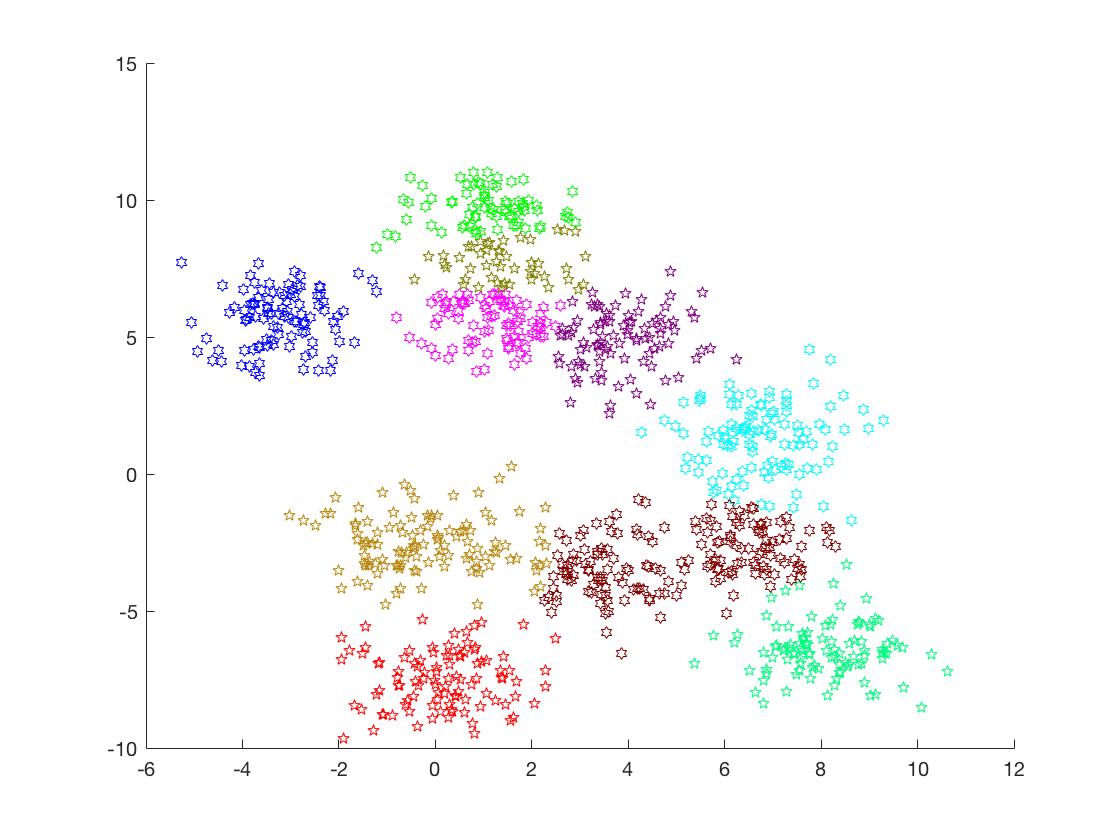




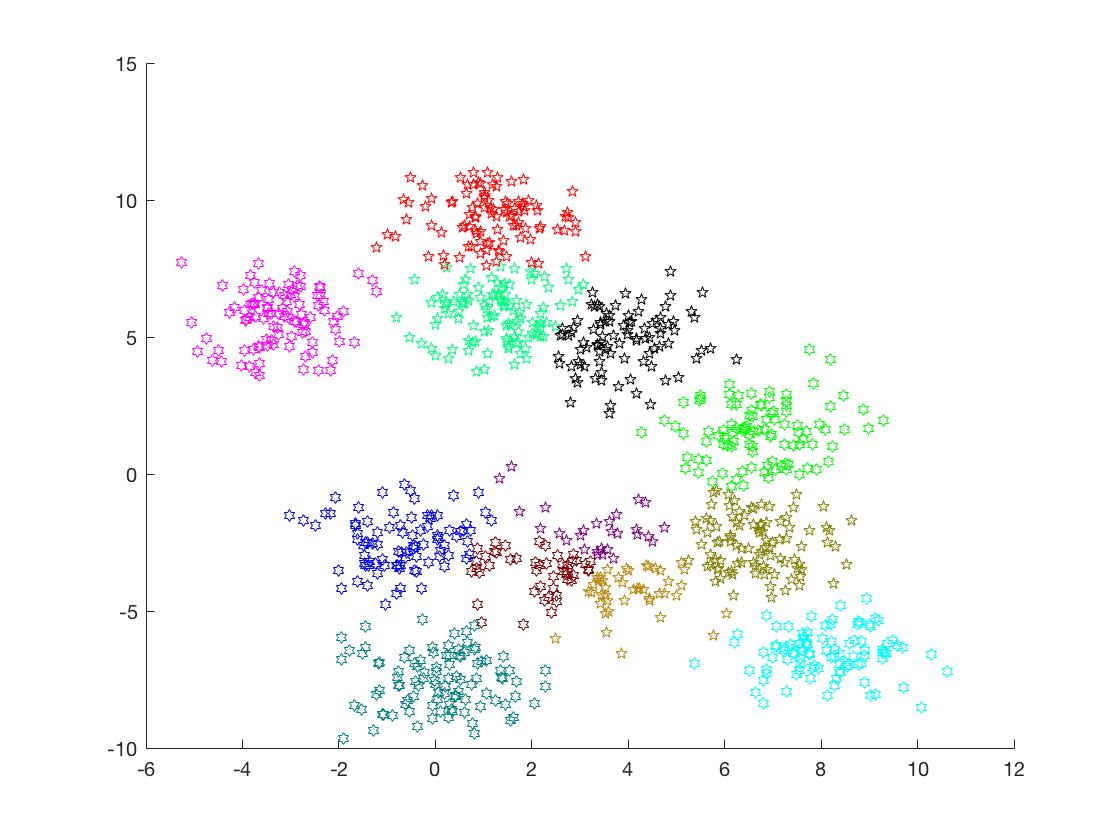
500точек, k = 8 и k = 6 соответственно

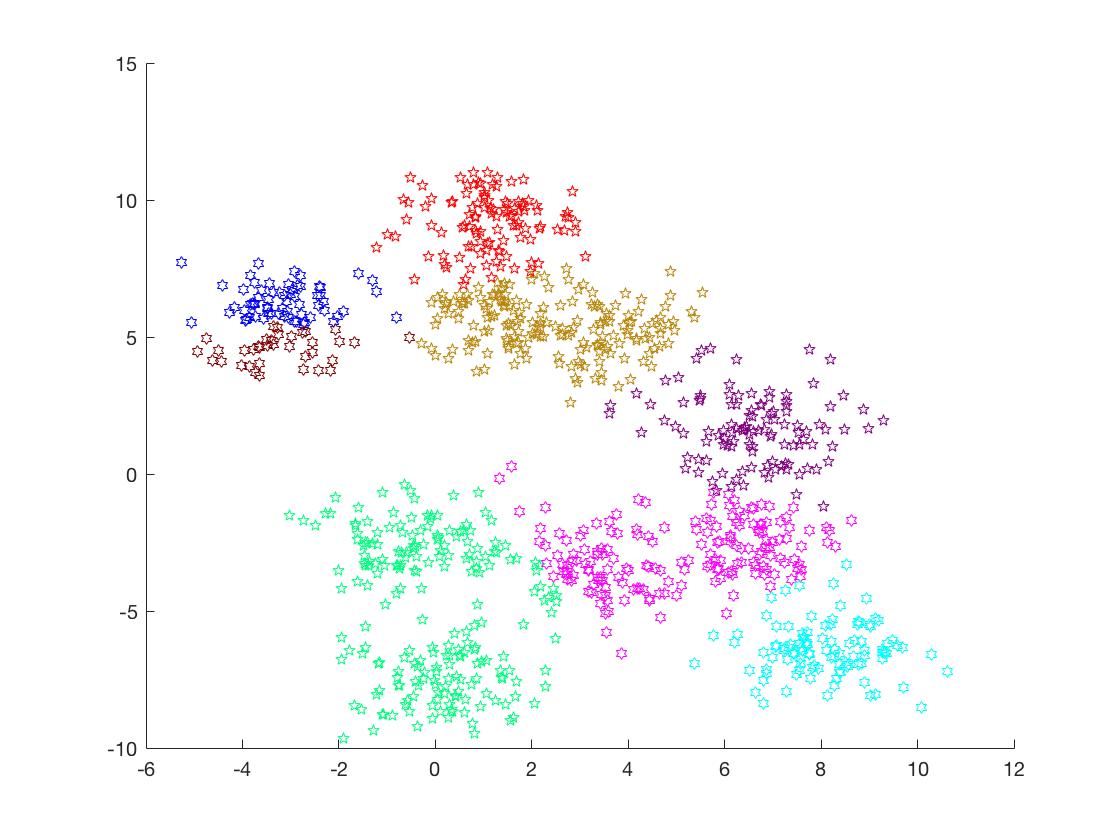


1000 точек, графики SS(k) и WSS(k) соответственно

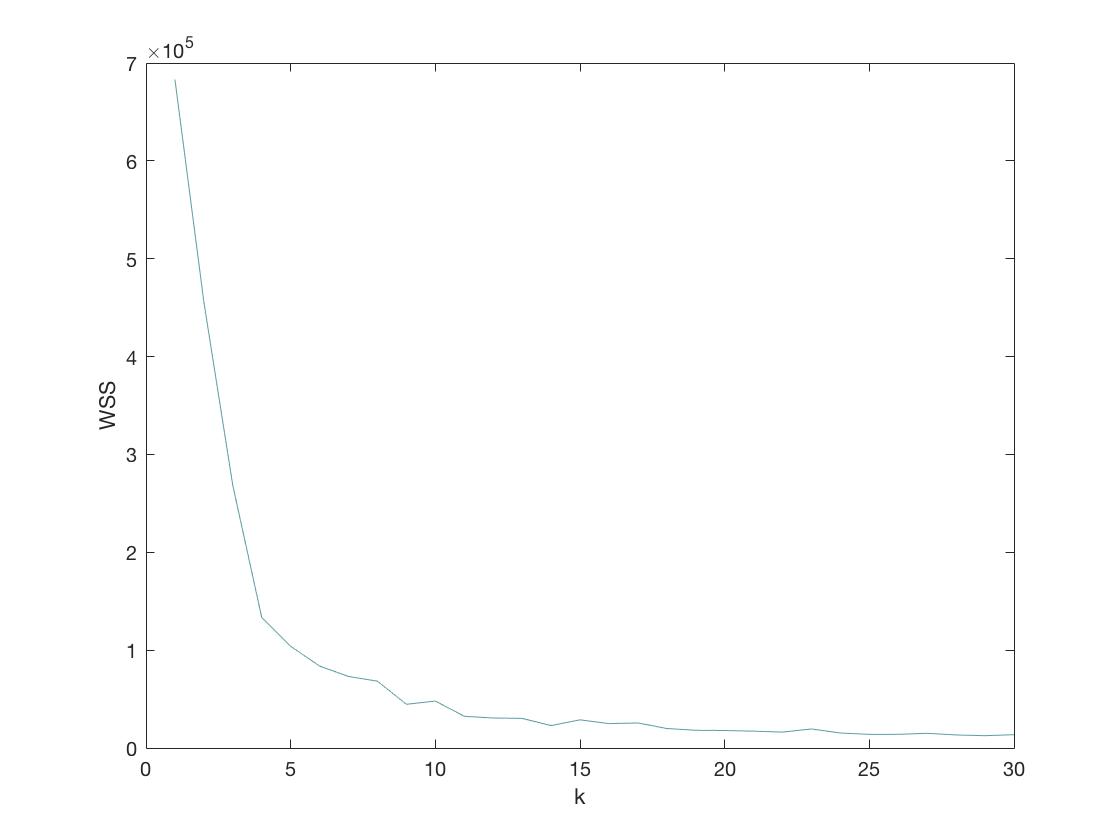
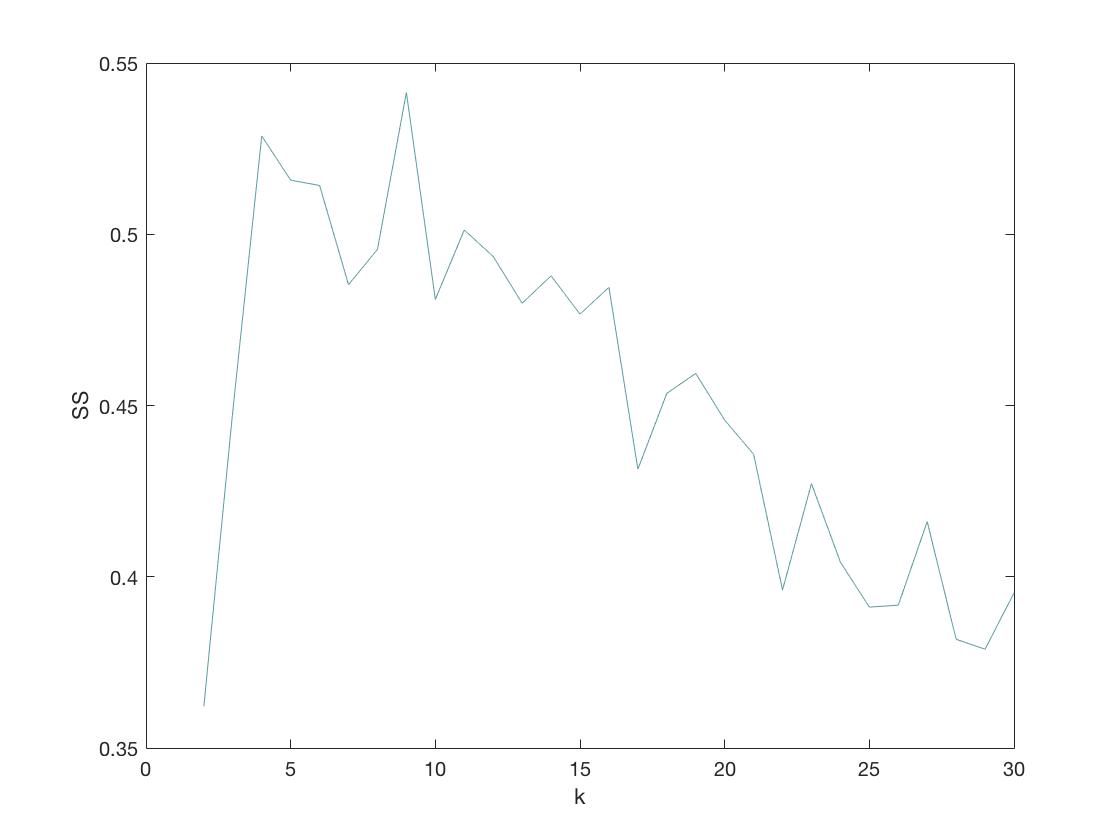


500точек, k = 10 и k = 9 соответственно

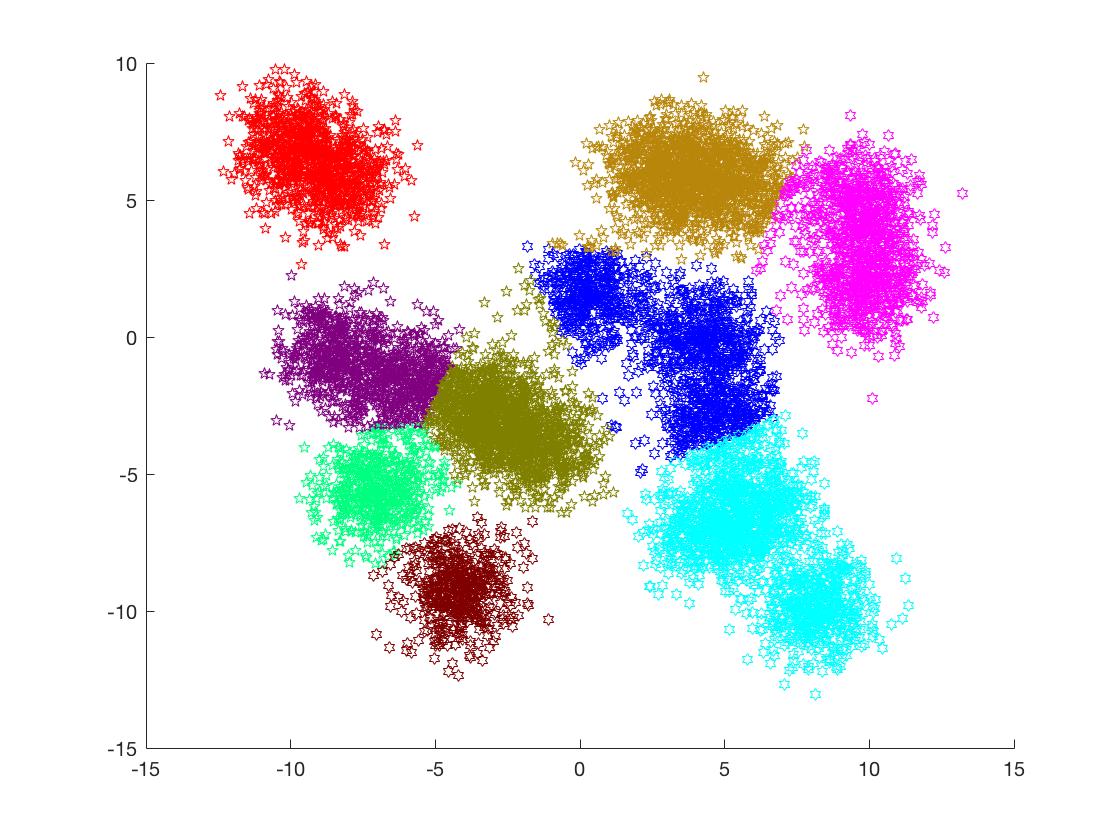


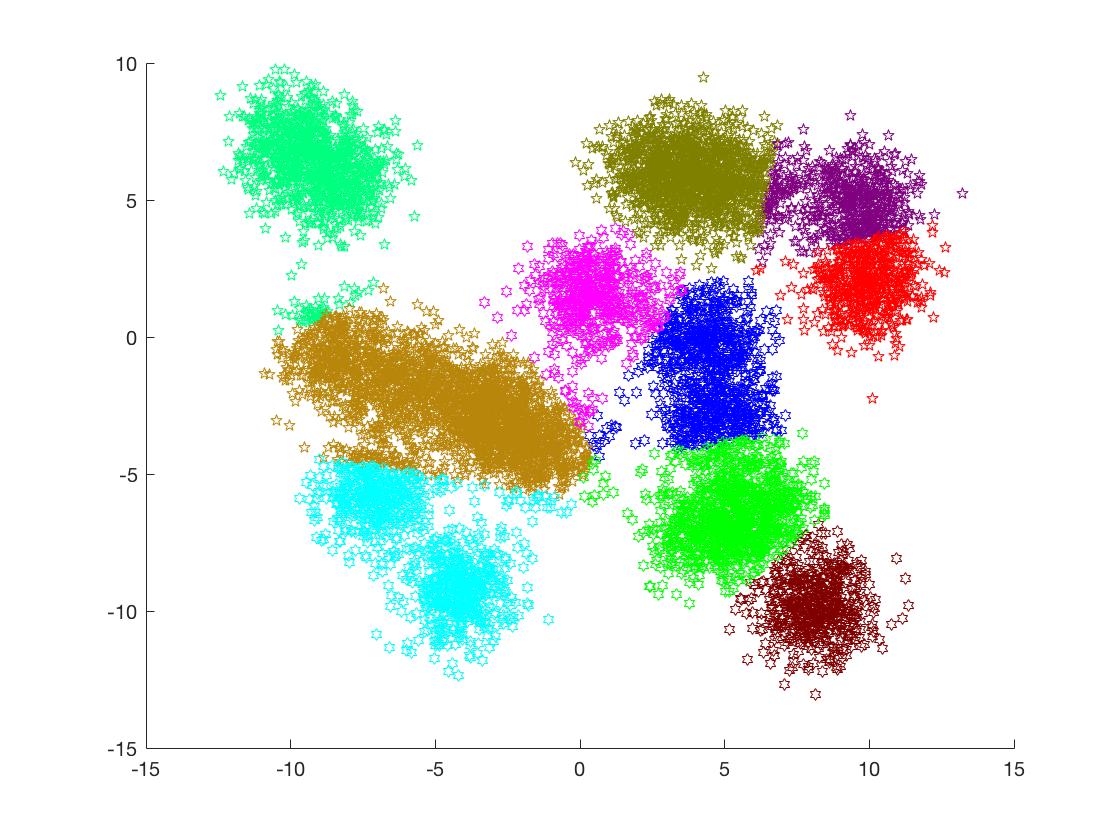


1000 точек, k = 12 и k = 8 соответсвенно



10000 точек, графики SS(k) и WSS(k) соответственно





10000 точек, k = 9 и k = 10 соответственно