ФГБОУ ВО

Уфимский государственный авиационный технический университет

Кафедра ВМиК

Отчет

по лабораторной работе №6

Тема: «Кластеризация»

Выполнили: ст. гр. ПРО-411

Букатина В.В. и Кабирова Л.Р.

Проверила: Харисова Э. А.

Уфа 2019

**Цель:** приобрести навыки работы с системой статистического анализа данных R.

**Задачи:**

1. Разбейте множество объектов из набора данных pluton на 3 кластера методом центров тяжести (kmeans). Сравните качество разбиения в зависимости от максимального числа итераций алгоритма.
2. Сгенерируйте набор данных в двумерном пространстве, состоящий из 3 кластеров, каждый из которых сильно «вытянут» вдоль одной из осей. Исследуйте качество кластеризации методом clara в зависимости от 1) использования стандартизации; 2) типа метрики. Объясните полученные результаты.
3. Постройте дендрограмму для набора данных votes.repub (число голосов, поданных за республиканцев на выборах с 1856 по 1976 год). Проинтерпретируйте полученный результат.

**Ход работы:**

**1) Разбейте множество объектов из набора данных pluton на 3 кластера методом центров тяжести (kmeans). Сравните качество разбиения в зависимости от максимального числа итераций алгоритма.**

Кластерный анализ— многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя.

Задача кластеризации заключается в разбиении выборки \{x^{(i)}, i=\overline{1,N}\} на непересекающиеся подмножества таким образом, чтобы схожие точки (обычно близкие в некоторой метрике) попали в одно подмножество (кластер), а точки из разных кластеров сильно друг от друга отличались (были далеки). Для решения данной задачи в системе статистического анализа R реализован метод центров тяжести (k-means). Он является одним из наиболее популярных алгоритмов кластеризации на настоящий момент.

K-means – это алгоритм разделительной кластеризации, основанный на разбиении множества элементов векторного пространства на заранее определенное число кластеров k. Алгоритм представляет собой итерационную процедуру, в которой выполняются следующие шаги.

Выбирается число кластеров k. Из исходного множества данных случайным образом выбираются k записей, которые будут служить начальными центрами кластеров. Для каждой записи исходной выборки определяется ближайший к ней центр кластера. При этом записи, «притянутые» определенным центром, образуют начальные кластеры.

Вычисляются центроиды – центры тяжести кластеров. Каждый центроид – это вектор, элементы которого представляют собой средние значения признаков, вычисленные по всем записям кластера. Затем центр кластера смещается в его центроид. Затем 3-й и 4-й шаги итеративно повторяются. Очевидно, что на каждой итерации происходит изменение границ кластеров и смещение их центров. В результате минимизируется расстояние между элементами внутри кластеров.

Остановка алгоритма производится тогда, когда границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор записей. На практике алгоритм обычно находит набор стабильных кластеров за несколько десятков итераций.

Преимуществом алгоритма являются быстрота и простота реализации. К его недостаткам можно отнести неопределенность выбора начальных центров кластеров, а также то, что число кластеров должно быть задано изначально, что может потребовать некоторой априорной информации об исходных данных.

Набор данных pluton состоит из 45 строк и 4 столбца, содержащие проценты изотопного состава 45 плутониевых партий.

Разобьем набор данных pluton на 3 кластера. Для сравнения качества разбиения используем различное количество итераций в экспериментах.

График разбиения набора данных pluton на кластеры для 1 итерации предоставлен на рис. 1.

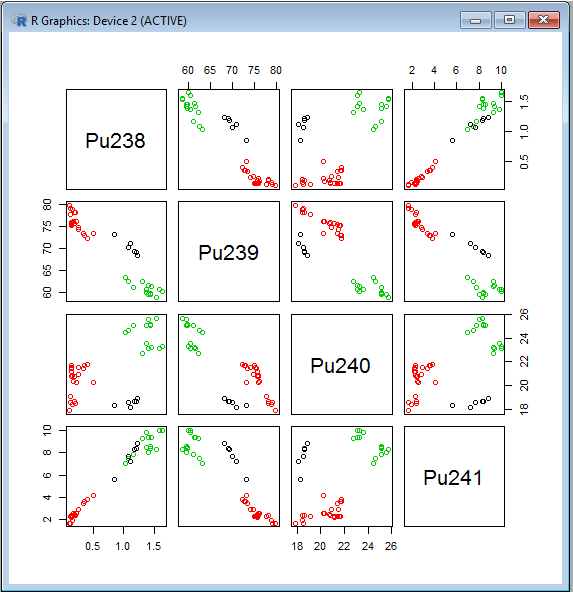


Рисунок 1 - *график кластеризации набора данных pluton методом k-средних (1 итерация)*

График разбиения набора данных pluton на кластеры для 5 итераций предоставлен на рис. 2.

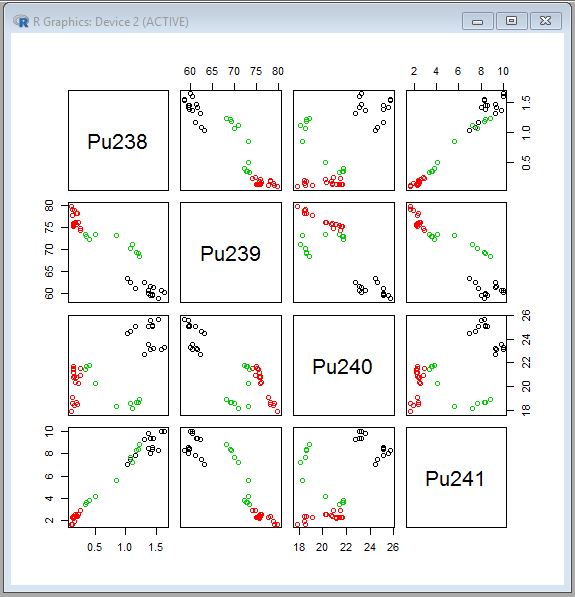


Рисунок 2 - *график кластеризации набора данных pluton методом k-средних (5 итераций)*

Графики при 1 и 5 итерациях значительно отличаются. Попробуем увеличить количество итераций до 10. График разбиения набора данных pluton на кластеры для 10 итераций предоставлен на рис. 3.

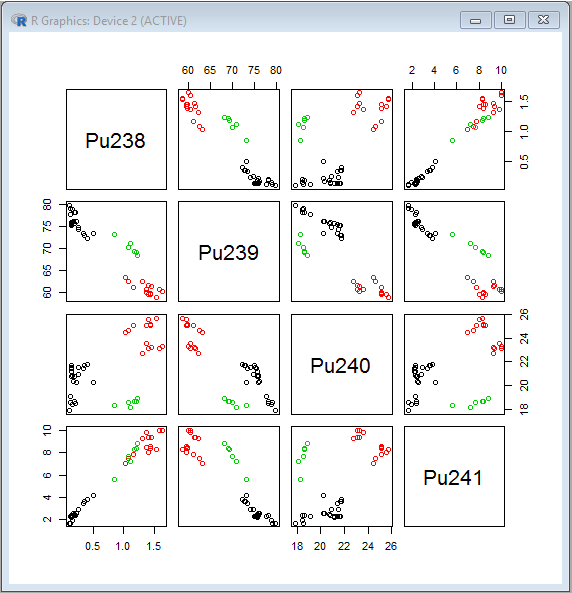


Рисунок 3 - *график кластеризации набора данных pluton методом k-средних (10 итераций)*

График с 10 количеством итераций немного отличается от графика с 5 количеством итераций. Попробуем еще увеличить количество итераций алгоритма до 100. График разбиения набора данных pluton на кластеры для 100 итераций предоставлен на рис. 4

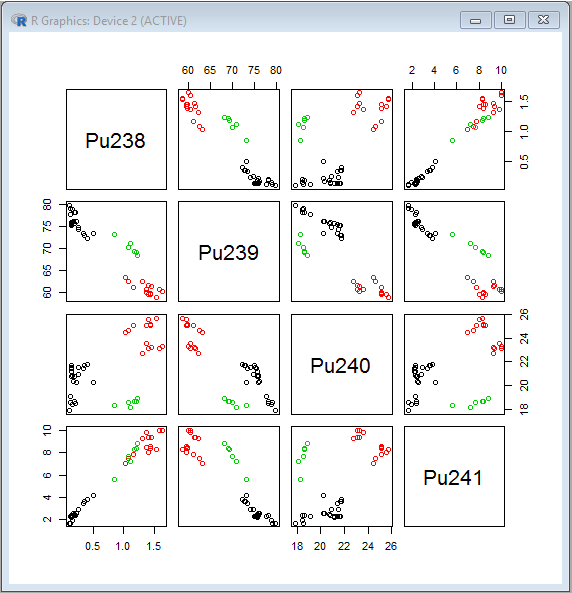
. 

Рисунок 4 - *график кластеризации набора данных pluton методом k-средних (100 итераций)*

Можем заметить, что график 100 итераций не отличается от графика 10 итераций алгоритма *k-средних.*

Итак, опытным путем было выяснено, что при увеличении количества итераций после 10 график кластеризации перестал изменяться. Это означает, что 10 итераций достаточно для качественного и точного разделения набора данных pluton методом k-средних.

Распишем результаты кластеризации для лучшего случая (10 итераций).

Объем элементов в кластерах:

https://lh6.googleusercontent.com/qy9qnqHgDSW6yKftKAKixydQktsDmBDJdNf-bjOscTo9cV2byir6eic195sqWx-2nyRCupVQYNwP5_T7izFD6sEssNKTCpXhlxixg5OQulYxILfjb9Uz9eWeXBRI2gFrd2QPeoQulaN0duBXHg

Рисунок 5 – *объем элементов в кластерах (10 итераций)*

Построим таблицу, отображающую интерпретацию полученных данных.

*Таблица 1 – объем элементов в кластерах*

|  |  |
| --- | --- |
| Номер кластера | Объем элементов в кластере |
| 1 | 15 |
| 2 | 12 |
| 3 | 18 |

Далее подсчитаем долю межкластерной дисперсии. Результаты расчетов представлены на рис. 6.

https://lh4.googleusercontent.com/oa48W-pRwBTm5WHVarmyndHKZXggTAaIM9TB_lLvtYL5sQAkNgUPP3dP1OJcTa3rYf1QfO_uOGQ2bU1HMgajYO5ssmdxaVZf_IoX4Nz-JCk7LI2bG5GWCpQCW5NWWTRGkeAVcSgj3azgoByHBQ

Рисунок 6 - доля межкластерной дисперсии *(10 итераций)*

Затем выведем на экран кластерный вектор. Он представлен на рис. 7.

https://lh5.googleusercontent.com/cGBlaFifv8nmtWFFcEI8GMA_fZrWk1Ay3olzad9MlLrZ73aBBbEAuV1GuXqo7iJ4n1IkJIwlTkYmFkOdcmcWNnwyOF_Tbyhr6qOB44Q-tmdddpVTFNrmRigvjRiO5Y6y7wKmVuvziwRMNr8gnA

Рисунок 7 - кластерный вектор *(10 итераций)*

Таким образом, мы получили всю необходимую информацию о результатах кластеризации для лучшего случая с 10 итерациями.

**Листинг выполнения задания 1:**

library(cluster) #подключение библиотеки cluster  
data(pluton) #загружаем набор данных pluton

cl <- kmeans(pluton, 3,1) #кластеризация набора данных pluton

plot(pluton, col = cl$cluster) #построение графика кластеризации набора данных pluton методом k-средних

cl#информация о результатах кластеризации

**2) Сгенерируйте набор данных в двумерном пространстве, состоящий из 3 кластеров, каждый из которых сильно «вытянут» вдоль одной из осей. Исследуйте качество кластеризации методом clara в зависимости от 1) использования стандартизации; 2) типа метрики. Объясните полученные результаты.**

Сперва генерируем исходный набор данных для анализа следующим образом (нормальным распределением rnorm):

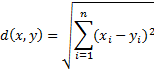
x <- rbind(cbind(rnorm(10000,0,10), rnorm(10000,10,0.1)), cbind(rnorm(10000,0,20), rnorm(10000,20,0.1)), cbind(rnorm(10000,15,0.2), rnorm(10000,8,10)))

Теперь разделим эти данные на три кластера методом кластеризации clara. Это довольно известная модификация PAM (Partitioning Around Medoids). Из исходного графа случайным образом выбирается подмножество вершин, и кластеризуется подграф, образованный этими вершинами. Затем (в предположении связности графа) оставшиеся вершины просто распределяются по ближайшим медоидам из подграфа. Вся суть clara состоит в последовательном прогоне алгоритма на разных подмножествах вершин и выборе наиболее оптимального из результатов. За счет этого предполагается компенсировать ущерб от исключения части информации при каждом отдельном прогоне, а также избежать застревания в локальном минимуме.

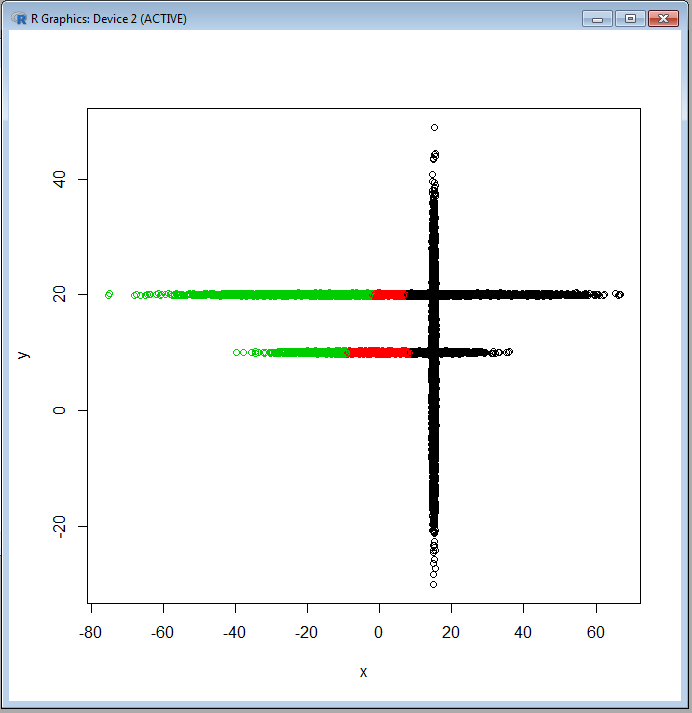
Clara содержит в себе несколько аргументов, в частости метрики и стандартизации. Для вычисления расстояния между объектами используются различные меры сходства (меры подобия), называемые также метриками или функциями расстояний. Наиболее популярная мера сходства - евклидово расстояние.

Евклидово расстояние является геометрическим расстоянием в многомерном пространстве.

Евклидово расстояние между точками x и y в n-мерном пространстве вычисляется по следующей формуле:



Для придания больших весов более отдаленным друг от друга объектам можем воспользоваться квадратом евклидова расстояния путем возведения в квадрат стандартного евклидова расстояния.



*Рисунок 8 – кластеризация методом clara без стандартизации с метрикой "euclidean" (metric = "euclidean")*

Стандартизация (standardization) или нормирование (normalization), приводит значения всех преобразованных переменных к единому диапазону значений путем выражения через отношение этих значений к некой величине, отражающей определенные свойства конкретного признака.

Стандартизация – это расчет нескольких сравниваемых совокупностей в целях исключения влияния структур на величину изучаемого показателя и приведения данных к сопоставимому виду.

Стандартизация показаний позволяет сравнить показатели, полученные испытуемым с таковыми в генеральной совокупности. В данном случае стандартизированные показатели мы получаем с помощью линейного преобразования первичных показателей (сырых данных).

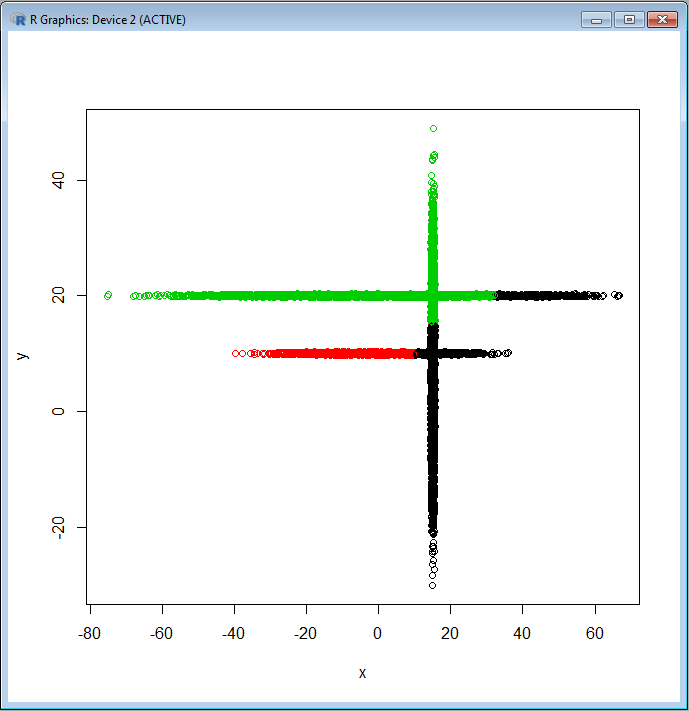
В этом случае показатели называются Z-стандартными и вычисляются по формуле:

https://lh5.googleusercontent.com/cv7xLgnUOFewh3v28tnVjDDzlvhWo2_welUyEg_lbP7es6tUSppe1k6lDIq6NxGBx08m6PWUNbmDpWkzSGgvWAW4qDZRee-jDBI4TGh2M-1JQ1ZGFmUYMA8uHfu6B_UshNepZQbl, где

Xi- индивидуальный балл каждого испытуемого по всему тесту,

https://lh3.googleusercontent.com/3fIcvO8hXrReo8oRg1GEpJScIf0Lb1kpniD2QDP85nYmTGy3beElOfsnfx3JonFJW6PHBmDnxcOuHYXyzIvQy7bk7jZbB3Awoc8ZdhWMiT7-DSmP83ZpmjJ3blZ-5WOX46hjwsMt- среднее арифметическое оценок по всему тесту всех испытуемых,

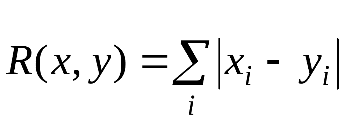
* = Sx - стандартное отклонение индивидуальных оценок всех испытуемых выборки.



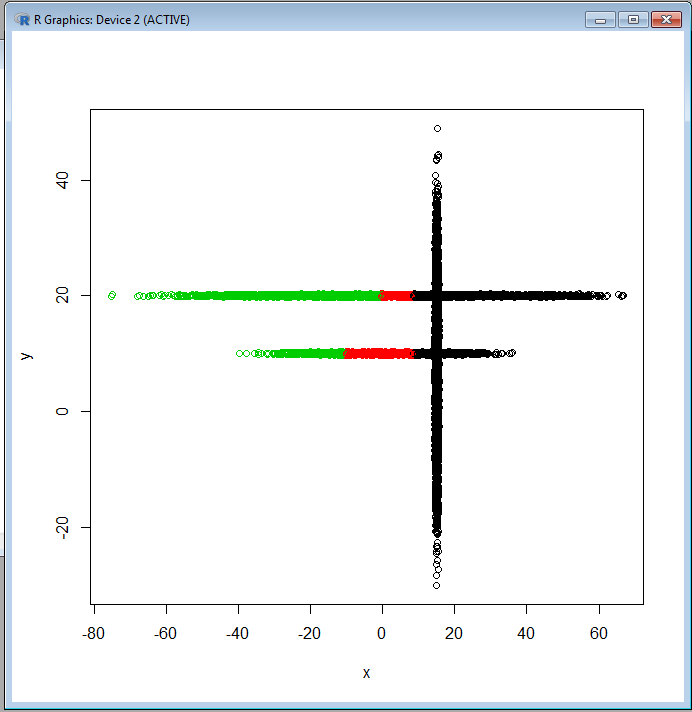
*Рисунок 9 – кластеризация методом clara со стандартизацией с метрикой "euclidean" (stand = TRUE, metric = "euclidean")*

Манхэттенское расстояние (расстояние городских кварталов), также называемое "хэмминговым" или "сити-блок" расстоянием. Это расстояние рассчитывается как среднее разностей по координатам. В большинстве случаев эта мера расстояния приводит к результатам, подобным расчетам расстояния евклида. Однако, для этой меры влияние отдельных выбросов меньше, чем при использовании евклидова расстояния, поскольку здесь координаты не возводятся в квадрат.

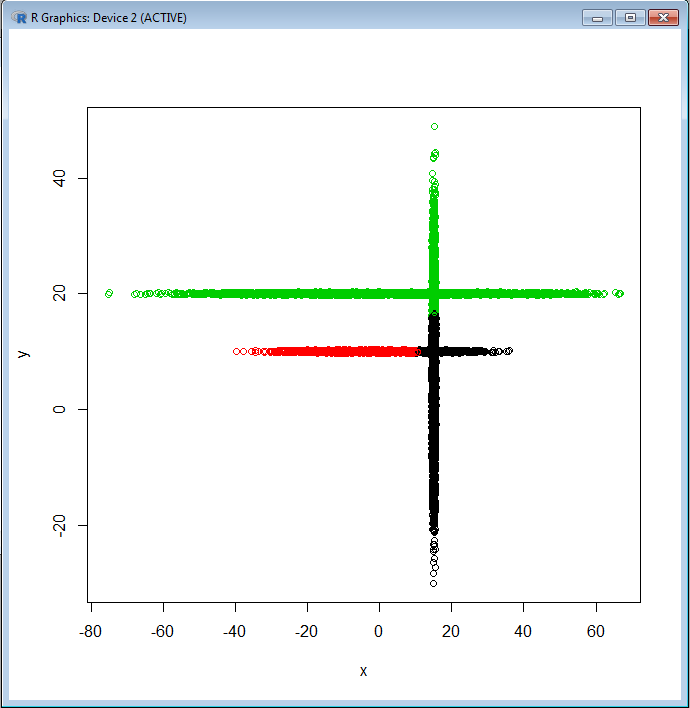
Манхэттенское расстояние вычисляется по формуле:



Теперь вычислим кластеризацию с манхэттенским расстоянием в R. Ниже предоставлены рисунки вычислений.



*Рисунок 10 – кластеризация методом clara без стандартизации с метрикой "manhattan" (metric = "manhattan")*



*Рисунок 11 – кластеризация методом clara со стандартизацией с метрикой "manhattan" (stand = TRUE, metric = "manhattan")*

Теперь построим таблицу, с помощью которой исследуем качество кластеризации методом clara в зависимости от 1) использования стандартизации; 2) типа метрики. Для того, чтобы это сделать, необходимо вычислить количество ошибок кластеризации. Встроенной функции в R для этого нет, так что напишем свою:

num\_of\_e <- function(x) {

sum = 0

for(i in 1:length(x)) {

if((floor(i/10000)+1)!=x[i]) {

sum = sum + 1

}

}

sum - 1

}

Далее подсчитаем количество ошибок, занесем их в табл. 2 и проанализируем получившиеся данные.

Таблица 2 - Количество ошибок кластеризации

|  |  |  |
| --- | --- | --- |
| Тип метрики | Стандартизация | Количество ошибок |
| "euclidean" | есть | 896 |
| "euclidean" | нет | 5472 |
| "manhattan" | есть | 794 |
| "manhattan" | нет | 5027 |

По итогам сравнения полученных данных можно сделать вывод, что методы стандартизации увеличивают точность кластеризации как в случае евклидового расстояния, так и в случае манхэттеновского расстояния. Результатом с наименьшим количеством ошибок кластеризации является кластеризация с типом метрики «манхэттеновское расстояние» и со стандартизацией.

**Листинг:**

x <- rbind(cbind(rnorm(10000,0,10), rnorm(10000,10,0.1)), cbind(rnorm(10000,0,20), rnorm(10000,20,0.1)), cbind(rnorm(10000,15,0.2), rnorm(10000,8,10))) #генерация данных для последующего анализа

clarax <- clara(x,3, stand=TRUE, metric = "euclidean") #clara со стандартизацией с метрикой "euclidean"

plot(x,col=clarax$clustering, xlab="x", ylab="y") #отрисовка для clara со стандартизацией с метрикой "euclidean"

clarax <- clara(x,3, stand=FALSE, metric = "euclidean") #clara без стандартизации с метрикой "euclidean"

plot(x,col=clarax$clustering, xlab="x", ylab="y") #отрисовка для clara без стандартизации с метрикой "euclidean"

clarax <- clara(x,3, stand=FALSE, metric = "manhattan") #clara без стандартизации с метрикой "manhattan"

plot(x,col=clarax$clustering, xlab="x", ylab="y") #отрисовка для clara без стандартизации с метрикой "manhattan"

clarax <- clara(x,3, stand=TRUE, metric = "manhattan") #clara со стандартизацией с метрикой "manhattan"

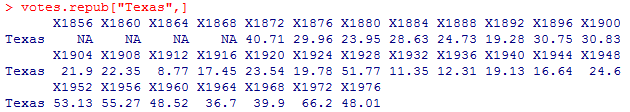
plot(x,col=clarax$clustering, xlab="x", ylab="y") #отрисовка для clara со стандартизацией с метрикой "manhattan"

num\_of\_e(clarax$clustering) #вычисляем ошибки кластеризации

**3) Постройте дендрограмму для набора данных votes.repub (число голосов, поданных за республиканцев на выборах с 1856 по 1976 год). Проинтерпретируйте полученный результат.**

Набор данных votes.repub - фрейм данных с процентами голосов, отданных кандидату-республиканцу на президентских выборах с 1856 по 1976.

Сначала для примера выведем распределение голосов по годам в штате Texas.



*Рисунок 12 – распределение голосов по годам в штате Texas*

Теперь построим график распределения голосов по годам в штате Texas.

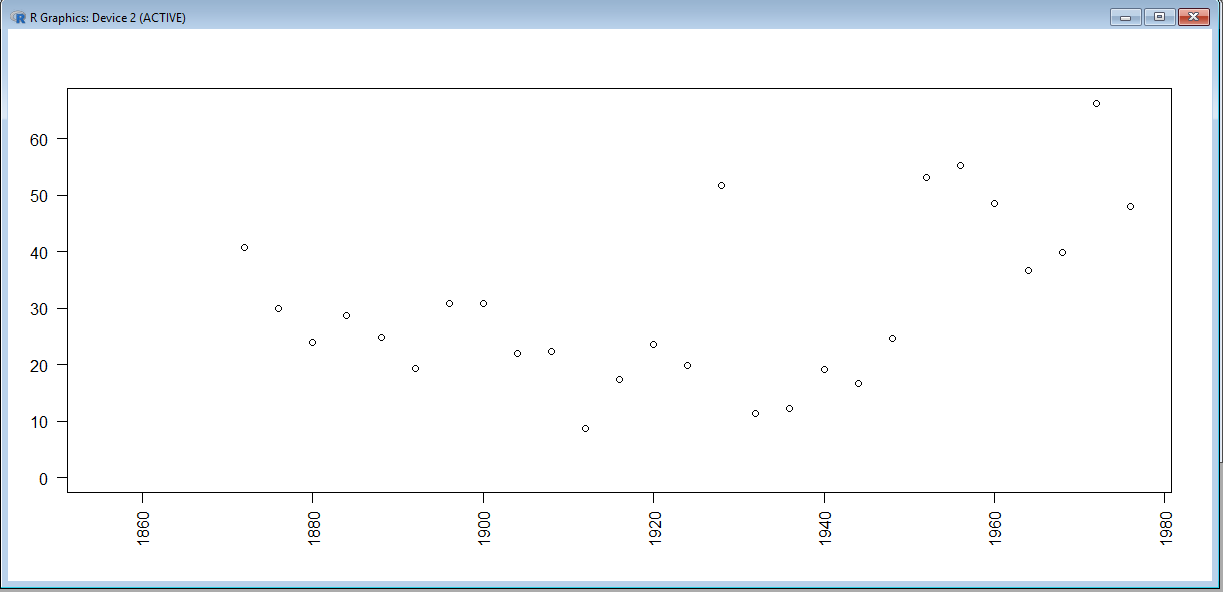
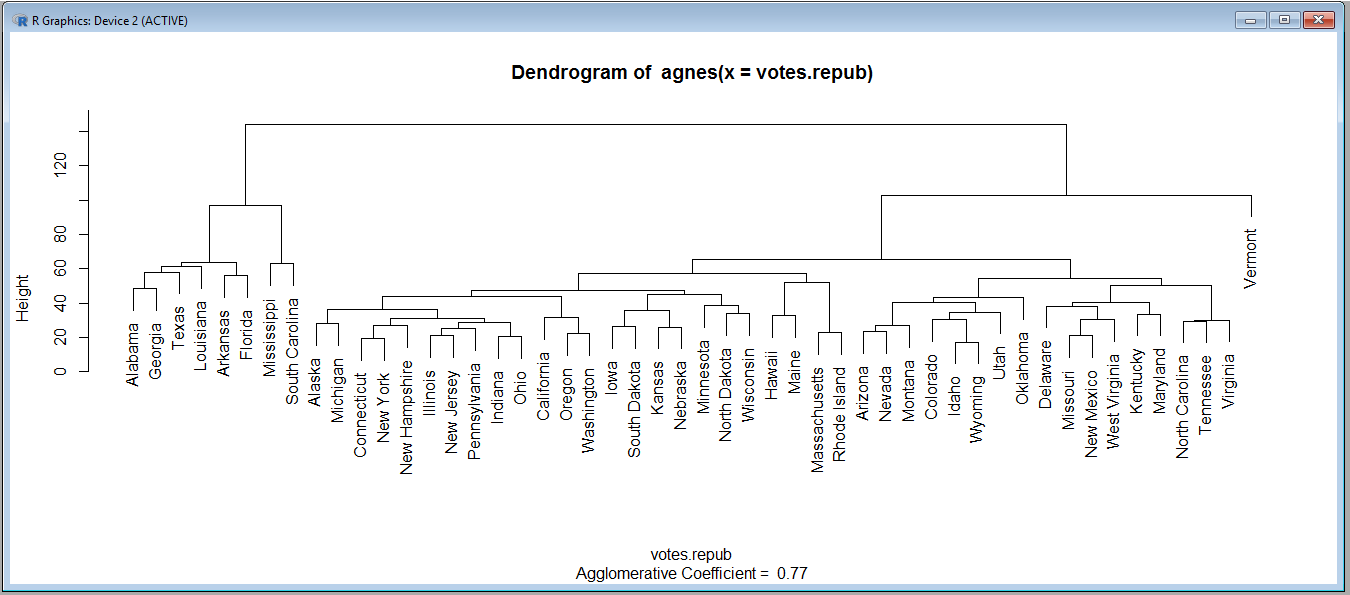


Рисунок 13 *– график распределения голосов по годам в штате Texas*

Затем построим дендрограмму для набора данных votes.repub. Дендрограмма - это диаграмма, представляющая дерево. Это схематическое представление часто используется в разных контекстах: в иерархической кластеризации это иллюстрирует расположение кластеров, произведенных соответствующими анализами. Дендрограмма показывает степень близости отдельных объектов и [кластеров](https://wiki.loginom.ru/articles/cluster.html), а также наглядно демонстрирует в графическом виде последовательность их объединения или разделения. Количество уровней дендрограммы соответствует числу шагов слияния или разделения кластеров. В левой части рисунка расположена шкала, на которой откладывается расстояние между объектами в пространстве признаков.



*Рисунок 14 – дендрограмма для набора данных votes.repub*

Ось Y является мерой близости отдельных точек данных или кластеров. Каждое соединение (слияние) двух кластеров представлено на диаграмме разбиением вертикальной линии на две вертикальные линии. Вертикальная позиция разделения, показанная короткой полосой, дает расстояние (различие) между двумя кластерами.

Из дендрограммы видно, что группируются объекты Alabama и Georgia образуя кластер (Alabama, Georgia) с минимальным расстоянием между объектами, примерно равным 50. Выше уровнем группируется кластер ((Alabama, Georgia), Texas) на отметке 58. Также образуют новый кластер (Louisiana, ((Alabama, Georgia), Texas)) - 59. (Arkansas, Florida) образуют кластер на уровне 57. ((Louisiana, ((Alabama, Georgia), Texas)), (Arkansas, Florida)) - 60. Mississippi и South Carolina образуют кластер (Mississippi, South Carolina) с равным расстоянием 60.

В свою очередь кластеры (Louisiana, ((Louisiana, ((Alabama, Georgia), Texas)), (Arkansas, Florida)) и (Mississippi, South Carolina) образуют новый кластер уже более высокого уровня на отметке 100.

По такому же принципу построена правая ветвь дендрограммы. Данный процесс продолжается до тех пор, пока все наблюдения не объединятся в один кластер.

По итогам дендрограммы можно сделать вывод, что наибольшее количество голосов за республиканцев принадлежит Vemont, они составляют 100, что делает Vemont лидером среди других городов. Vemont образует единый кластер с другой ветвью, которая включает в себя 41 город (на дендрограмме от Alaska до Verginia). В данной ветви можно выделить объединенный кластер (Hawai и Maine), у которого наиболее высокий показатель 40 среди кластеров срединной ветви. А также объединенный кластер ((Hawaii, Maine), (Massachusetts, Rhode Island)) - 50, который также является лидером среди своего подуровня в данной ветви. Наименьшее количество голосов за республиканцев принадлежит штатам Indiana и Ohio, по 15 голосов.

Рассмотрим левую большую ветвь, которая идет от самого высокого уровня кластера, включает в себя 8 городов (от Alabama до South Carolina). Лидерами в своей подветви являются Mississippi, South Carolina с показателями 60. Наименьшее количество голосов за республиканцев оказалось в штатах Alabama, Georgia по 58.

Итак, была построена дендрограмма для набора данных votes.repub. Также проведен анализ полученной дендрограммы, который показал, что большее количество голосов за республиканцев оказалось в штате Vemont, наименьшее - Indiana и Ohio по 15 голосов.

**Листинг выполнения задания 3:**

data(votes.repub) #загружаем набор данных votes.repub

plot(agnes(votes.repub)) #строим дендрограмму для набора данных votes.repub

**Вывод по лабораторной работе:**

При выполнении лабораторной работы были приобрести навыки работы с системой статистического анализа данных R. Также было выполнено задание для самостоятельной работы в соответствии с руководством по выполнению лабораторной работы и оформлен отчет о выполнении лабораторной работы в соответствии с требованиями к его оформлению.