**Отчёт по Лабораторной Работе № 3.**

Нам был предложен датасет на 2179 объектов и 4 признака.

Первое, что мы заметили это то, что для некоторых признаков есть как отрицательные, так и положительные значения, и было решено привести всё к положительным. Для этого мы к каждому значению широты прибавили 90, к каждому значению долготы 180. Теперь у нас широта и долгота лежат на отрезках [0;180] и [0;360] соответственно.

Второе, что мы заметили – то, что хоть у нас координаты теперь положительные, но они зависят от выбранной точки отсчёта, и расстояние между точками с координатами [45; 360] и [45;10] будет большое, хотя на самом деле точки находятся близко. Для этого мы “сделали сдвиг” точки отсчёта долготы на 180 градусов *(новая долгота = (старая долгота + 180) % 360).* 180 было выбрано из соображений, что меньше всего точек находится как раз в пределах этой долготы.

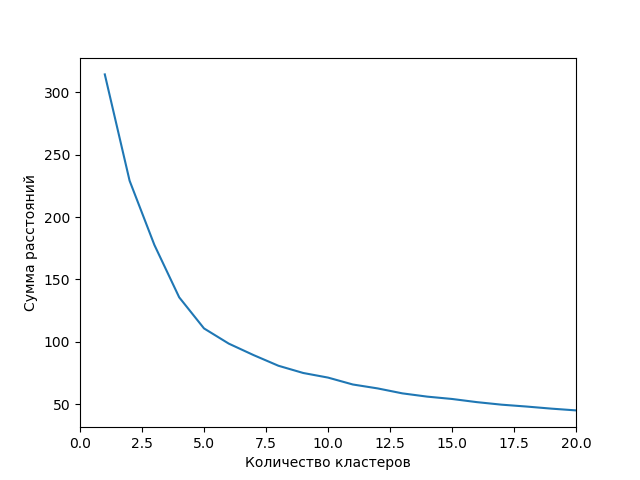
Второе, что было замечено – это сильный разброс между значениями признаков. Например, глубина землетрясения варьируется от 0 до 659 метров, координаты от -180 до 180, сила землетрясения [0;10]. Так как в дальнейшем нам нужно будет считать расстояния между объектами, то нужно чтобы значения всех признаков лежали в одном интервале, так как иначе у нас расстояние будет больше зависеть, например, от глубины, чем от силы, потому что значения глубины больше по модулю, чем значения силы. Мы выбрали интервал значений [0;1] и с помощью библиотеки *sklearn.preprocessing* и функции *MinMaxScaler()* привели значения к нему. В итоге теперь у нас все данные лежат на отрезке [0;1] и мы можем считать расстояние между объектами.

**K-means**

Для алгоритма k-means нам нужно знать количество кластеров, на которые мы хотим разбить наши объекты. Для этого используем метод *каменистой осыпи.* Для начала определим какое количество кластеров может быть максимально. Мы выбрали 20 изначально с рассчетом на то, что если что, возьмем больше. Выполним k-means поочередно для 1,2…20 кластеров. В итоге мы получим 20 массивов с номером кластера для каждого объекта датасета соответственно. Далее для каждого массива вычислим сумму расстояний каждого объекта до центра его кластера, используя функцию *inerthia\_*. Для 1-20 кластеров суммы расстояний получились соответственно:

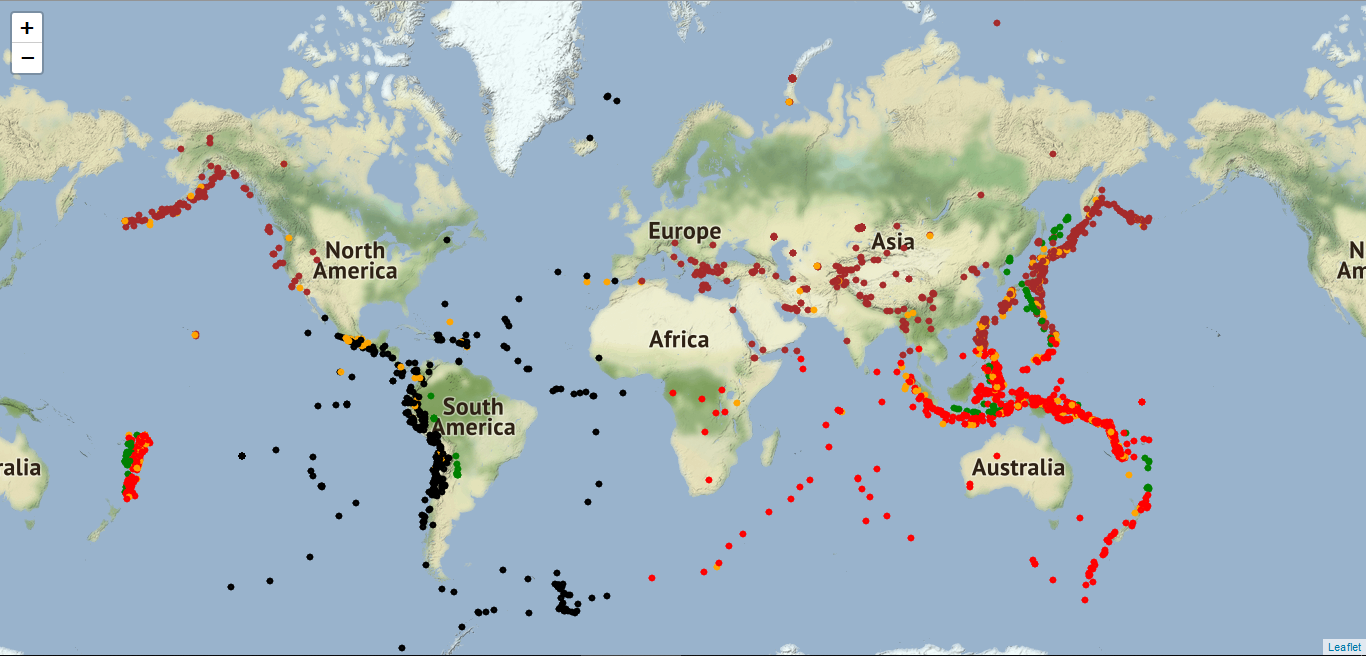
[314.2182091793016, 229.06171998160167, 177.60694720847462, 135.67134940707433, 110.77918943727248, 98.58812554212287, 89.4035561364693, 80.88027258909467, 75.0857104390629, 71.39619840766565, 65.87110345161985, 62.73375156266691, 58.756911915752795, 56.14379414486742, 54.270448960062836, 51.741127260066555, 49.69986772606795, 48.21135868622398, 46.53750251998942]

График каменистой осыпи:



Нам нужно найти оптимальное соотношение количества кластеров и значений расстояний, чтобы получить наилучшую кластеризацию. В данном случае целесообразно взять 5, т.к. во-первых, после 5, значения сумм убывают гораздо медленнее, во-вторых, при построении графика отчетливо видно все 5 кластеров и их можно логически интерпретировать.

С количеством кластеров определились, осталось только вернуть обратно сетку координат, визуализировать кластеры и дать им логическое обоснование. Т.к. у нас получилось пятимерное пространство, а построить его на трехмерной плоскости не представляется возможным, будем строить точки относительно их координат (долгота, широта), а за принадлежность тому или иному кластеру будет отвечать цвет.

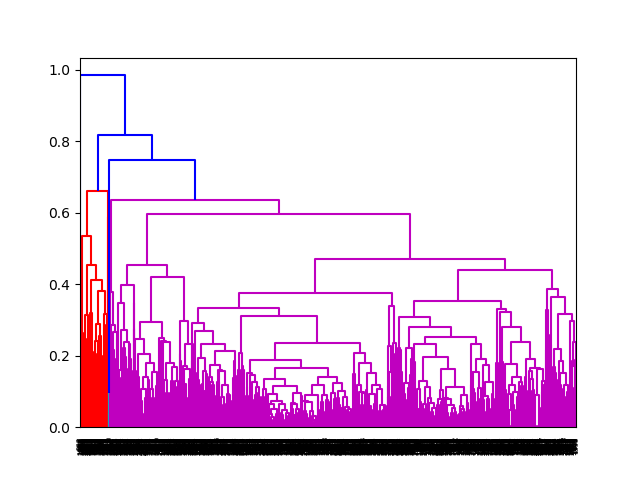


**TODO: описание картинки**

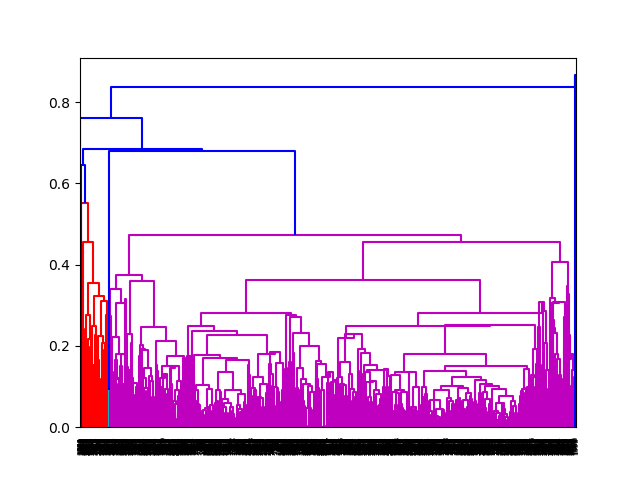
**Иерархическая кластеризация**

Для иерархической кластеризации есть множество методов и метрик расстояния. Мы попробовали все методы и несколько метрик. Представим несколько из них в виде дендрограм и постараемся выбрать оптимальную. Но сначала проведем те же преобразования с данными, что и в k-means (сдвинем данные по долготе и приведем к отрезку [0;1] ). Далее будем использовать пакет *scipy.cluster.hierarchy.*

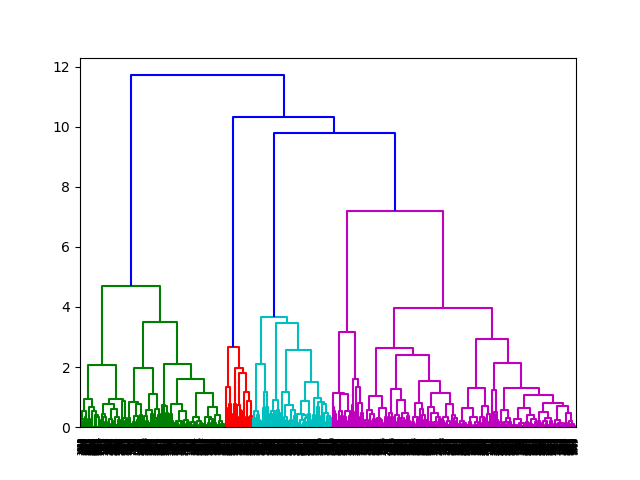
Для начала возьмем евклидово расстояние и метод average



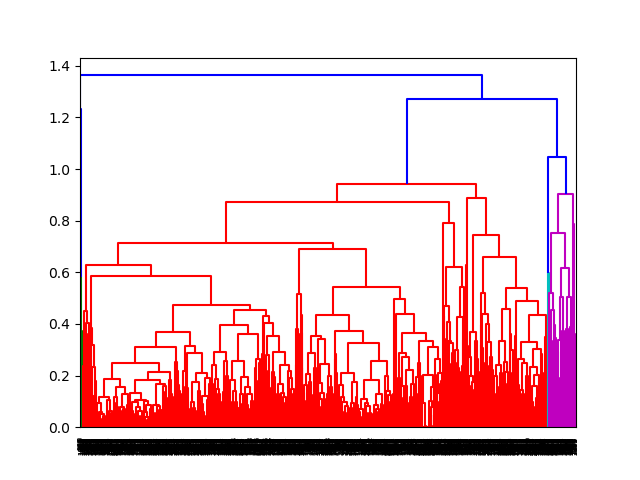
Евклидово расстояние и метод centroid



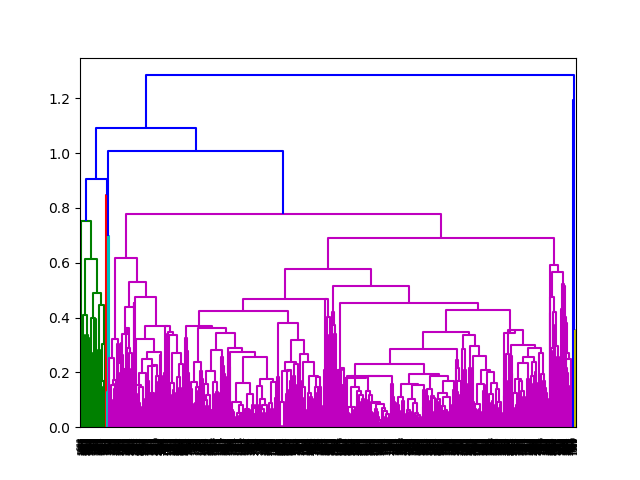
Евклидово расстояние и метод ward



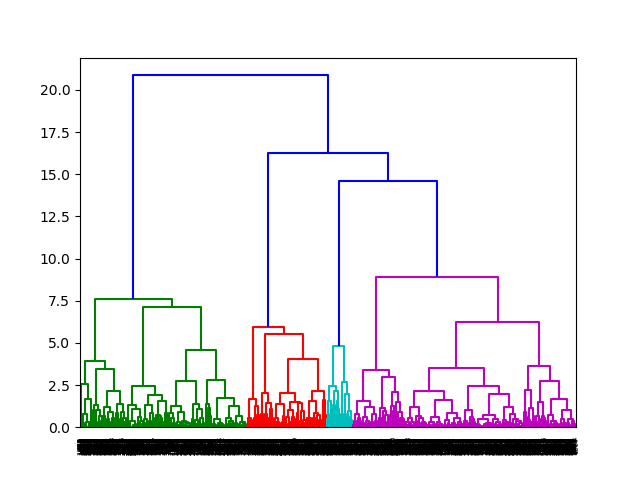
Манхэттенское расстояние и метод average

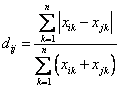


Манхэттенское расстояние и метод centroid

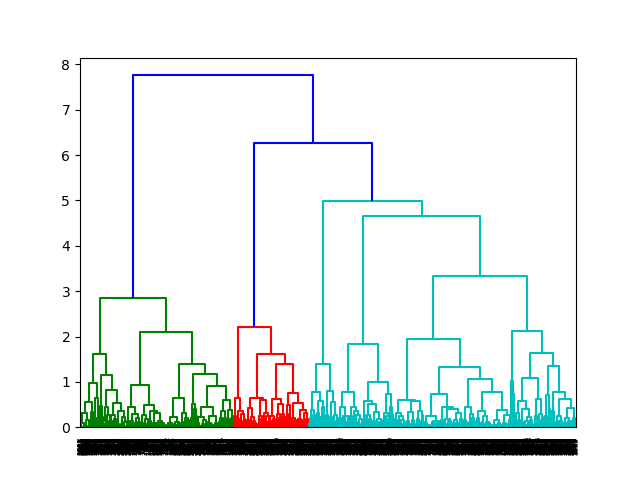


Манхэттенское расстояние и метод ward

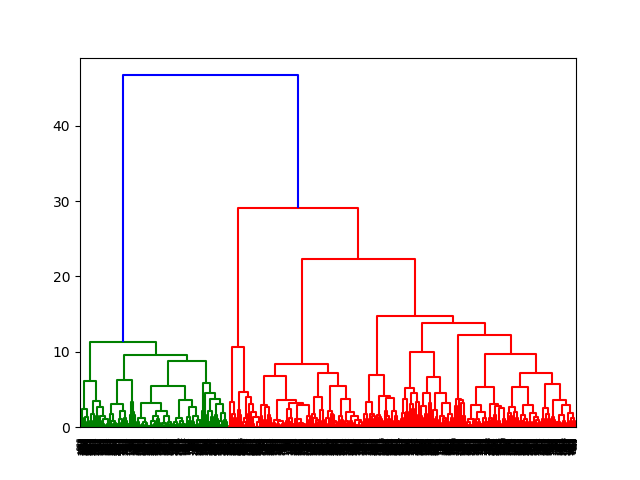


Мы заметили, что метод Уорда дает нам наилучшую кластеризацию, осталось выбрать метрику расстояния. Стандартные уже попробовали, теперь попробуем нестандартные. Возьмем метрику Bray–Curtis. Её формула . Она хорошо, когда все координаты положительные и лежат на отрезке [0;1] (как раз для нашего случая).

В итоге получаем вот такую дендрограму:



И расстояние Canberra. Оно тоже хорошо подходит для измерения расстояния между объектами, находящимися на отрезке [0;1] (само расстояние может быть и больше). Формула: Canberra Distance.



Итог: метод Уорда дал наилучший результат при кластеризаци, а за меру расстояния для дальнейшей работы мы выбрали метрику Bray–Curtis. Судя по дендрограме количество кластеров следует выбрать 6-7, мы взяли 6, т.к. посчитали его оптимальным.

В итоге получаем вот такую картину:

