Кластерный анализ

Анализ и визуализация многомерных данных с использованием R

М. Варфоломеева В. Хайтов А. Лянгузова

Кластерный анализ

- Методы построения деревьев
- Методы кластеризации на основании расстояний
- Примеры для демонстрации и для заданий
- Кластерный анализ в R
- Качество кластеризации:
  - кофенетическая корреляция
  - ширина силуэта
  - поддержка ветвей
- Сопоставление деревьев: танглграммы
- Неиерархические методы кластеризации:
  - K-means
  - C-means
  - DBSCAN

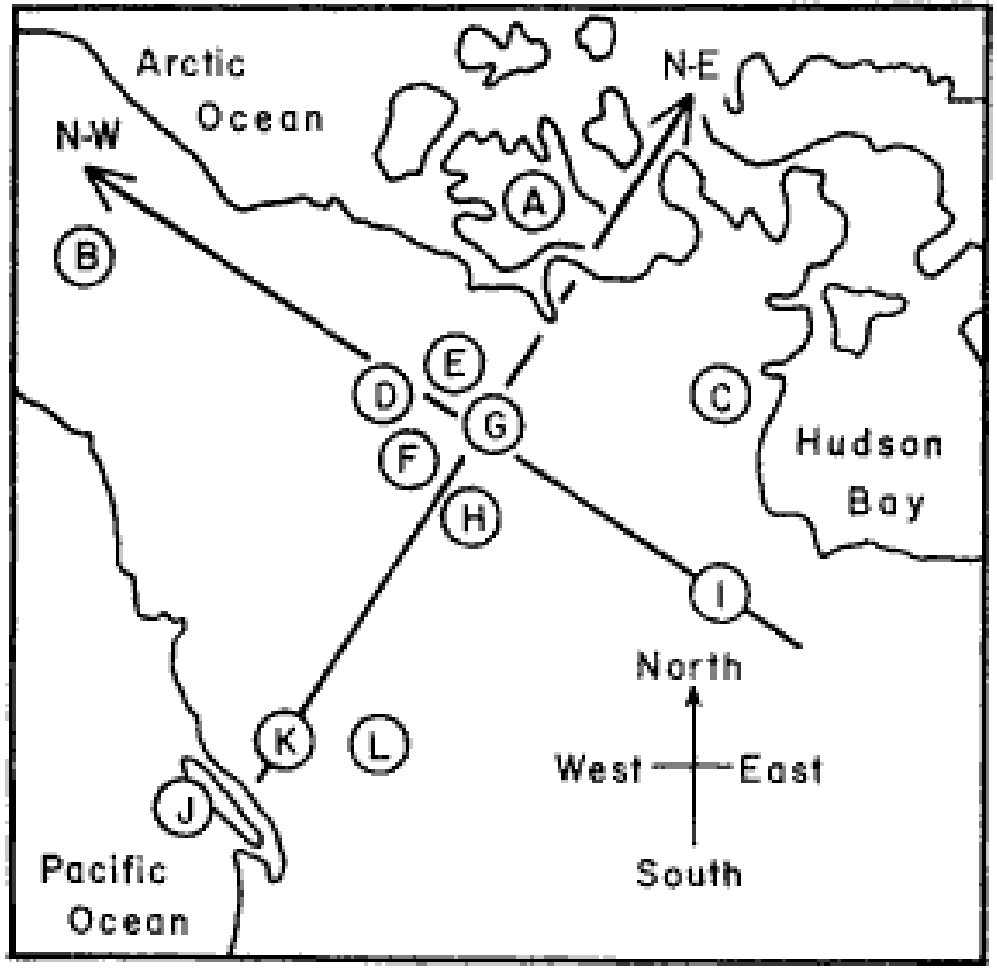
Вы сможете

- Выбирать подходящий метод агрегации (алгоритм кластеризации)
- Строить дендрограммы
- Оценивать качество кластеризации (кофенетическая корреляция, ширина силуэта, поддержка ветвей)
- Сопоставлять дендрограммы, полученные разными способами, при помощи танглграмм

Пример: Волки

Пример: Волки

Морфометрия черепов у волков в Скалистых горах и в Арктике (Jolicoeur, 1959)



Map from Jolicoeur 1959

Знакомимся с данными

```
dim(Wolves)
```

```
[1] 25 12
```

```
colnames(Wolves)
```

```
 [1] "group"    "location" "sex"      "x1"       "x2"       "x3"
 [7] "x4"       "x5"       "x6"       "x7"       "x8"       "x9"
```

```
head(rownames(Wolves))
```

```
[1] "rmm1" "rmm2" "rmm3" "rmm4" "rmm5" "rmm6"
```

```
any(is.na(Wolves))
```

```
[1] FALSE
```

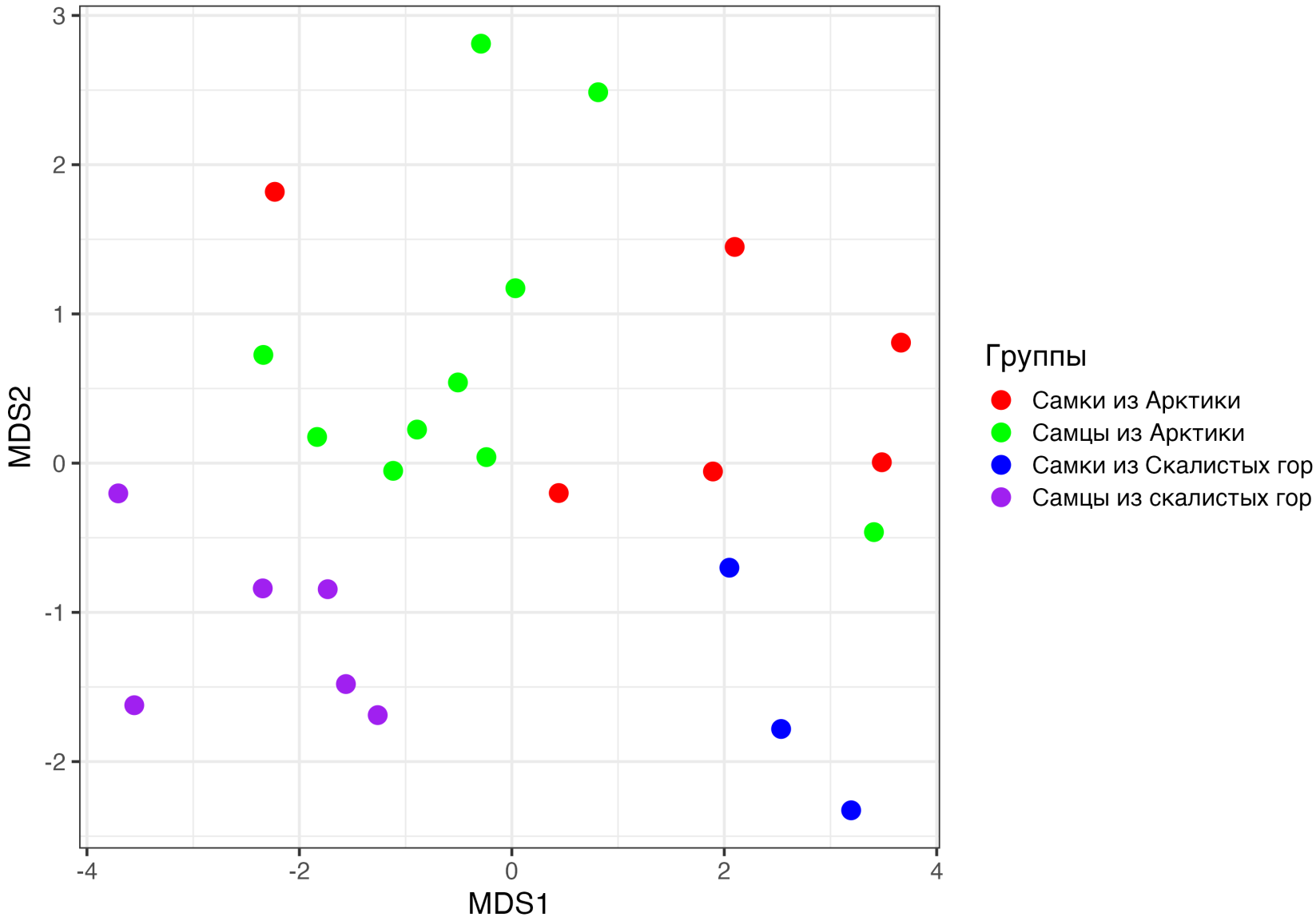
```
table(Wolves$group)
```

```
ar:f ar:m rm:f rm:m
   6   10    3    6
```

Задание

1. Постройте ординацию nMDS данных.
2. Оцените качество ординации.
3. Обоснуйте выбор коэффициента.
4. Раскрасьте точки на ординации волков в зависимости от географического происхождения (`group`).

Решение


MDS2
MDS1
3
2
1
0
-1
-2
-4
-2
0
2
4
Группы
Самки из Арктики
Самцы из Арктики
Самки из Скалистых гор
Самцы из скалистых гор

Решение

```
library(vegan)
library(ggplot2); theme_set(theme_bw(base_size = 16))
st_w <- scale(Wolves[, 4:ncol(Wolves)]) ## стандартизируем
ord_w <- metaMDS(comm = st_w, distance = "euclidean", autotransform =
↪  FALSE)
dfr_w <- data.frame(ord_w$points, Group = Wolves$group)
gg_w <- ggplot(dfr_w, aes(x = MDS1, y = MDS2)) +
  geom_point(aes(colour = Group)) +
  scale_color_manual(labels = c("Самки из Арктики", "Самцы из Арктики",
                                "Самки из Скалистых гор",
                                "Самцы из скалистых гор"),
                     values = c("red", "green", "blue", "purple")) +
  labs(colour = "Группы")
```

```
Run 0 stress 0.100972
Run 1 stress 0.1433766
Run 2 stress 0.100972
... Procrustes: rmse 1.463843e-06  max resid 4.480716e-06
... Similar to previous best
Run 3 stress 0.1267966
Run 4 stress 0.100972
... Procrustes: rmse 9.176626e-06  max resid 2.906529e-05
... Similar to previous best
Run 5 stress 0.100972
... New best solution
... Procrustes: rmse 5.45332e-06  max resid 1.826026e-05
... Similar to previous best
Run 6 stress 0.1406304
Run 7 stress 0.1380797
Run 8 stress 0.1354586
```

Методы кластеризации

Иерархические методы

- методы построения деревьев (о них следующие слайды)

Неиерархические методы

- метод K-средних (K-means clustering)
- метод нечёткой кластеризации C-средних (C-means clustering, fuzzy clustering)
- Основанная на плотности пространственная кластеризация для приложений с шумами (Density-based spatial clustering of applications with noise, DBSCAN)

Какие бывают методы построения деревьев?

Методы кластеризации на основании расстояний (о них сегодня)

- Метод ближайшего соседа
- Метод отдалённого соседа
- Метод среднегруппового расстояния
- Метод Варда
- и т.д. и т.п.

Методы кластеризации на основании признаков

- Метод максимальной бережливости
- Метод максимального правдоподобия

И это еще далеко не всё!

Методы кластеризации на основании расстояний

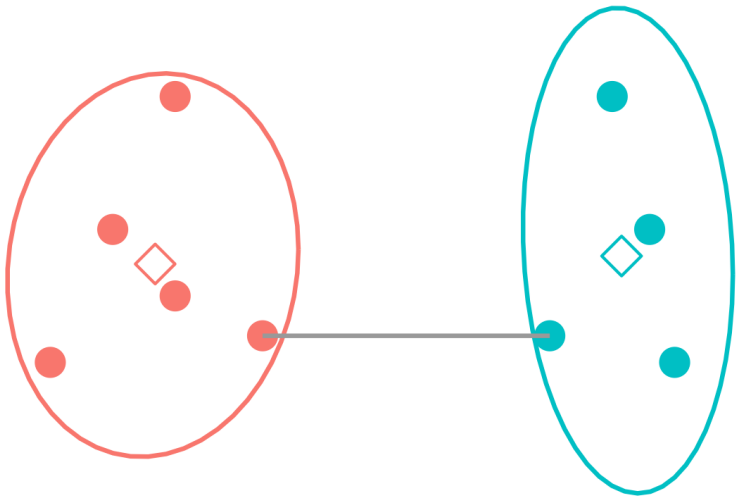
Этапы кластеризации

Результат кластеризации зависит от

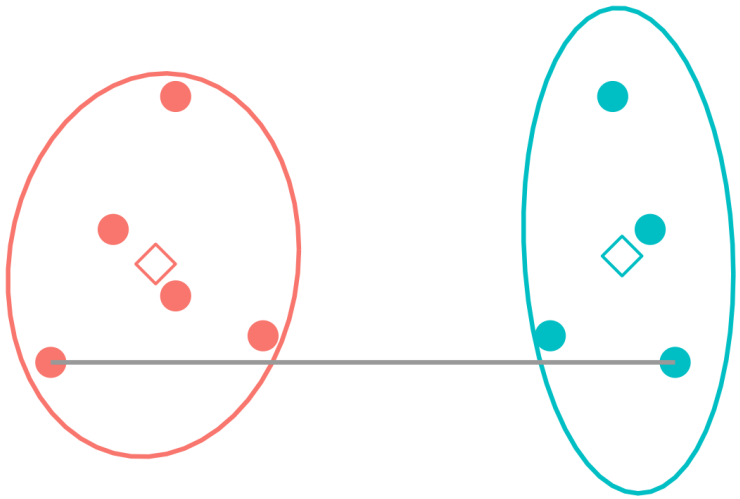
- выбора признаков
- коэффициента сходства-различия
- от алгоритма кластеризации

Методы кластеризации

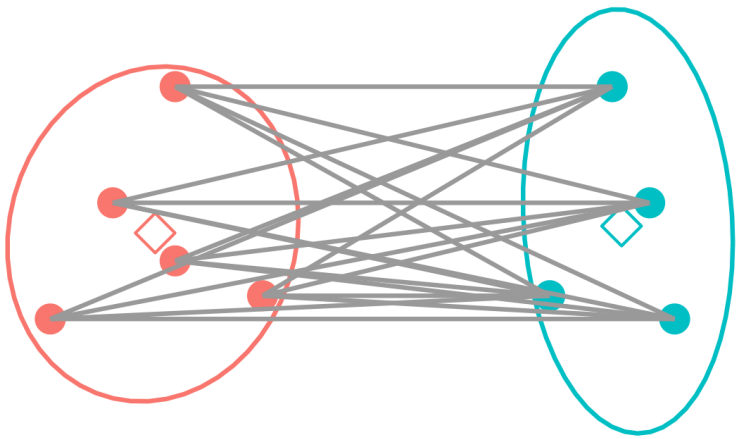
Метод ближайшего соседа



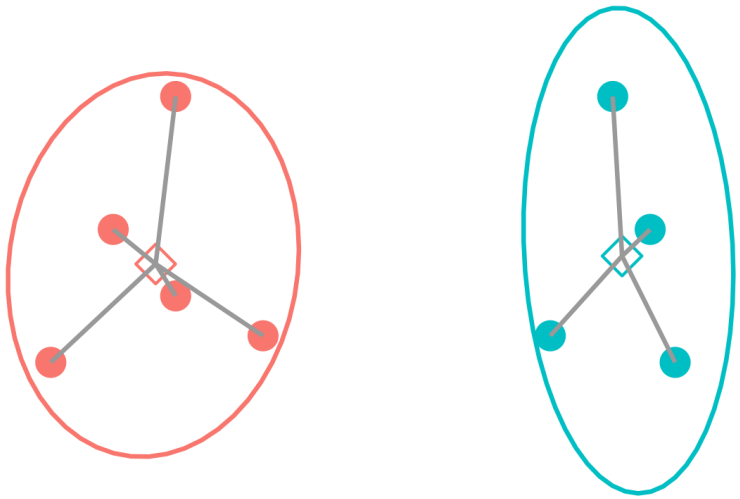
Метод отдаленного соседа



Метод среднегруппового расстояния



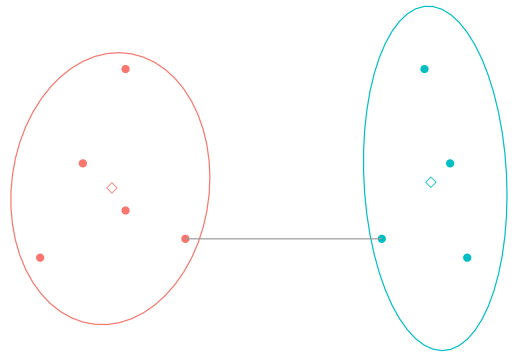
Метод Варда



Метод ближайшего соседа

= nearest neighbour = single linkage

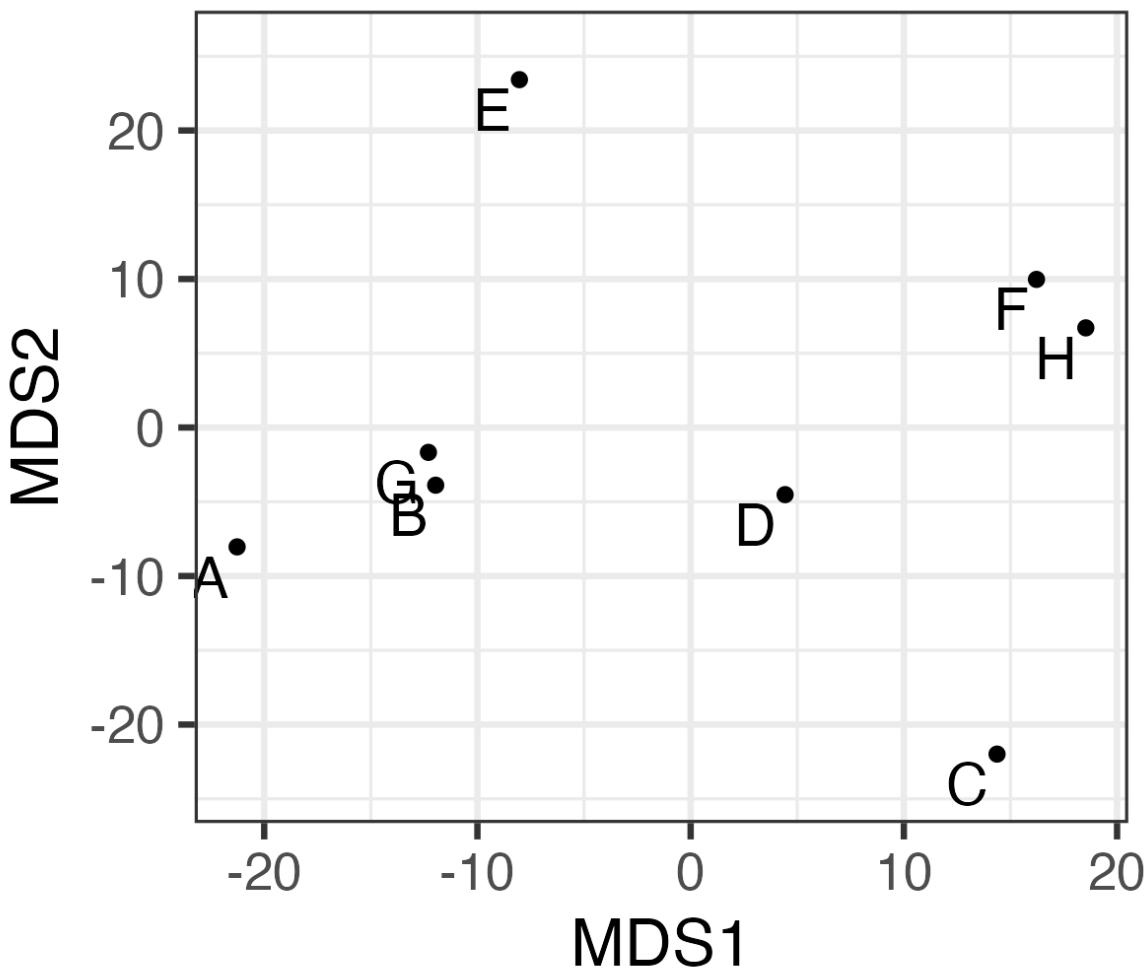
- к кластеру присоединяется ближайший к нему кластер/объект
- кластеры объединяются в один на расстоянии, которое равно расстоянию между ближайшими объектами этих кластеров



Особенности

- Может быть сложно интерпретировать, если нужны группы
  - объекты на дендрограмме часто не образуют четко разделенных групп
  - часто получаются цепочки кластеров (объекты присоединяются как бы по одному)
- Хорош для выявления градиентов

Как работает метод ближайшего соседа

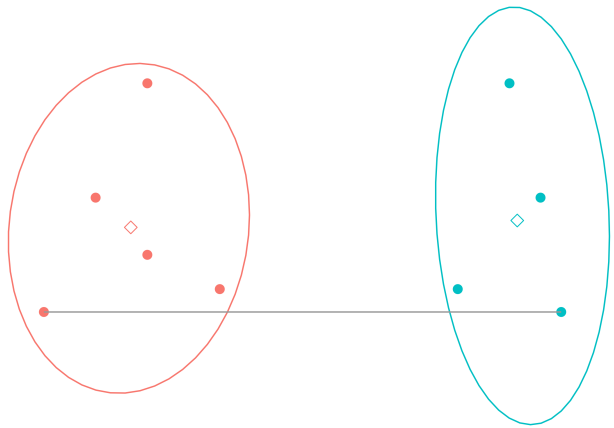


Анимация — в короткой .pptx презентации.

Метод отдаленного соседа

= furthest neighbour = complete linkage

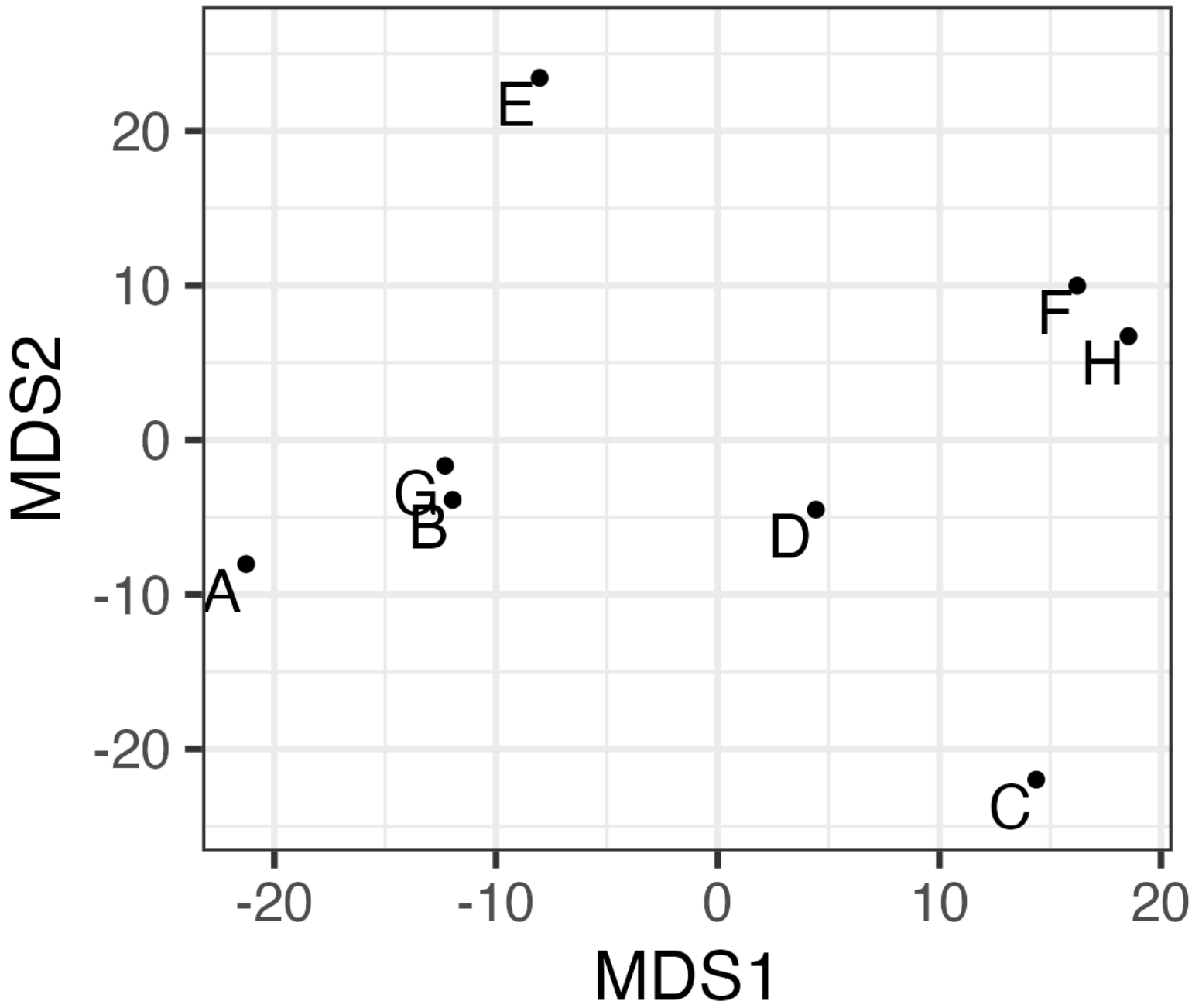
- к кластеру присоединяется отдаленный кластер/объект
- кластеры объединяются в один на расстоянии, которое равно расстоянию между самыми отдаленными объектами этих кластеров (следствие — чем более крупная группа, тем сложнее к ней присоединиться)



Особенности

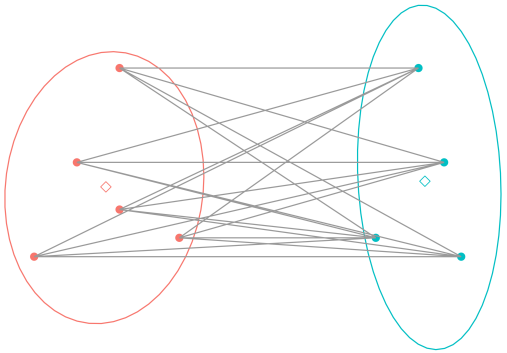
- На дендрограмме образуется много отдельных некрупных групп
- Хорош для поиска дискретных групп в данных

Как работает метод отдаленного соседа



Метод невзвешенного попарного среднего

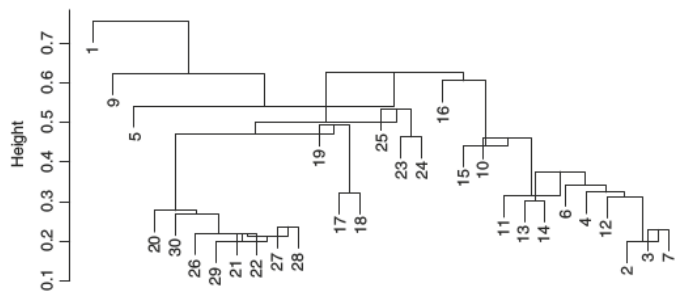
= UPGMA = Unweighted Pair Group Method with Arithmetic mean
кластеры объединяются в один на расстоянии, которое равно среднему значению всех возможных расстояний между объектами из разных кластеров.



Особенности

UPGMA и WPGMA иногда могут приводить к инверсиям на дендрограммах.

из Borcard et al., 2011



Инверсии на дендрограммах

UPGMA и WPGMA

UPGMA — **Unweighted** Pair Group Method with Arithmetic mean
Дистанция между кластерами рассчитывается:

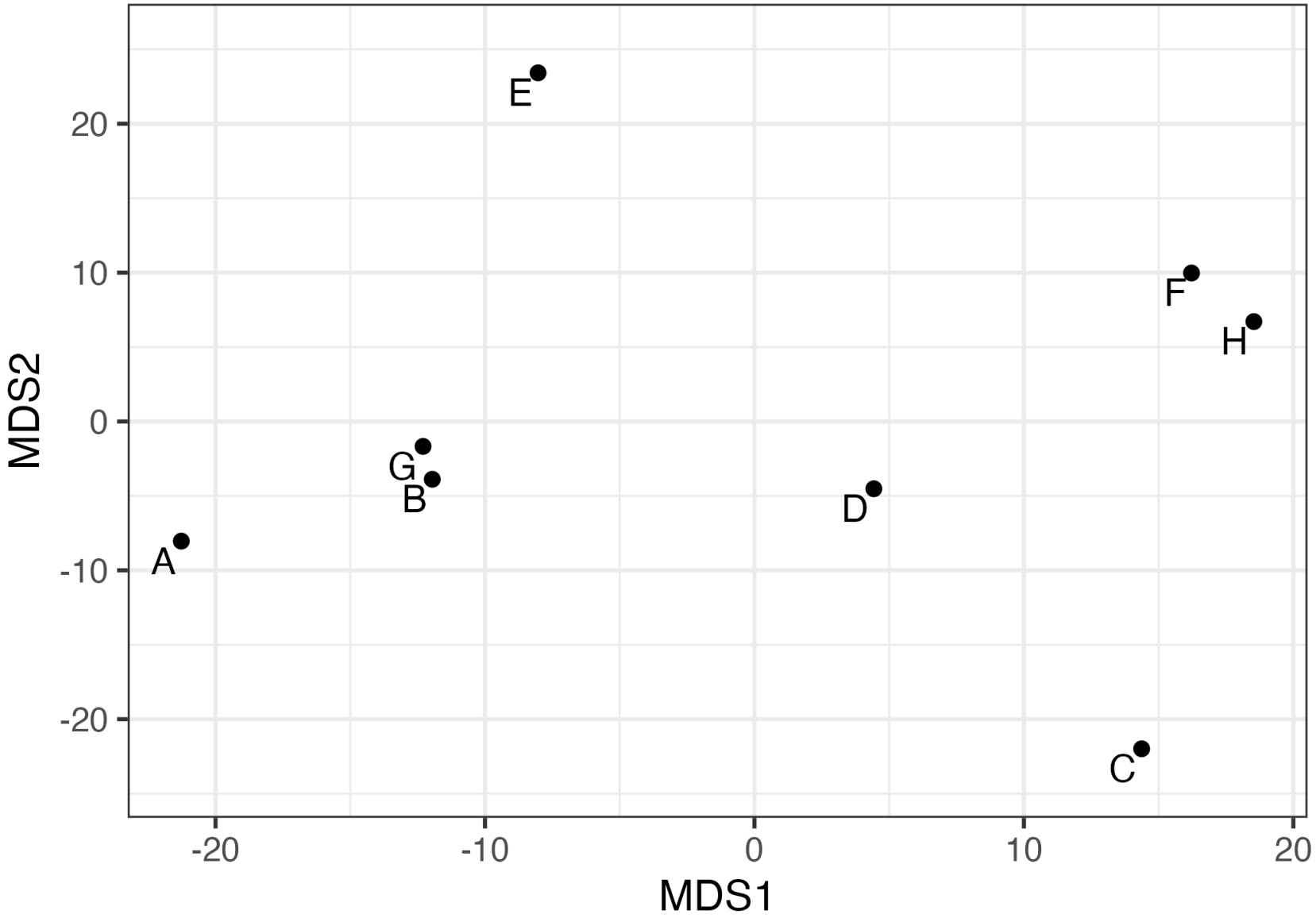
$$d_{(AB),C} = \frac{n_A \cdot d_{A,C} + n_B \cdot d_{B,C}}{n_A + n_B}$$

WPGMA — **Weighted** Pair Group Method with Arithmetic mean
Дистанция между кластерами рассчитывается:

$$d_{(AB),C} = \frac{d_{A,C} + d_{B,C}}{2}$$

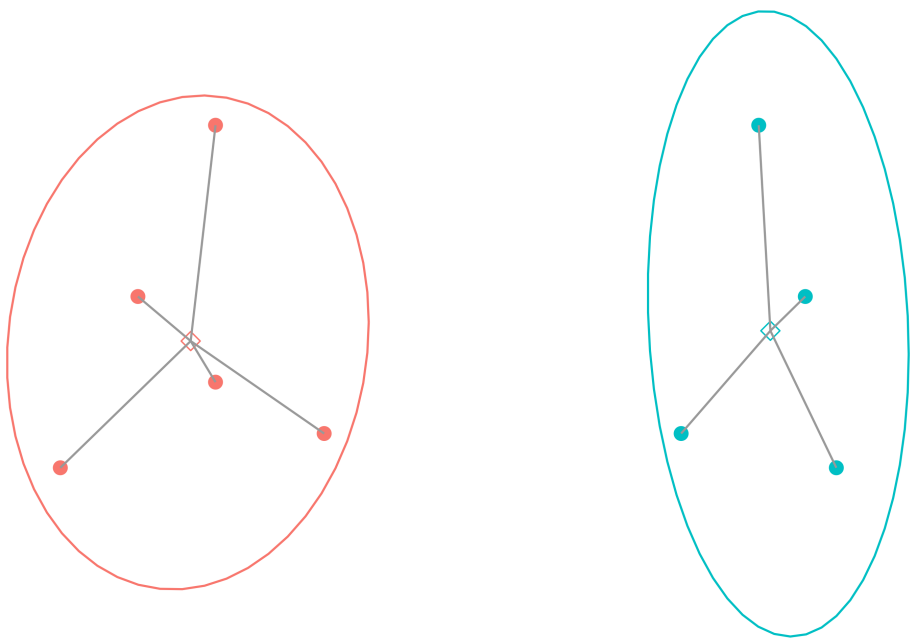
Взвешенный не потому, что задаются разные веса при кластеризации, а поскольку исходные расстояния оказывают неравное влияние на результат — побочный эффект игнорирования размера кластеров при расчете расстояния.

Как работает метод среднегруппового расстояния



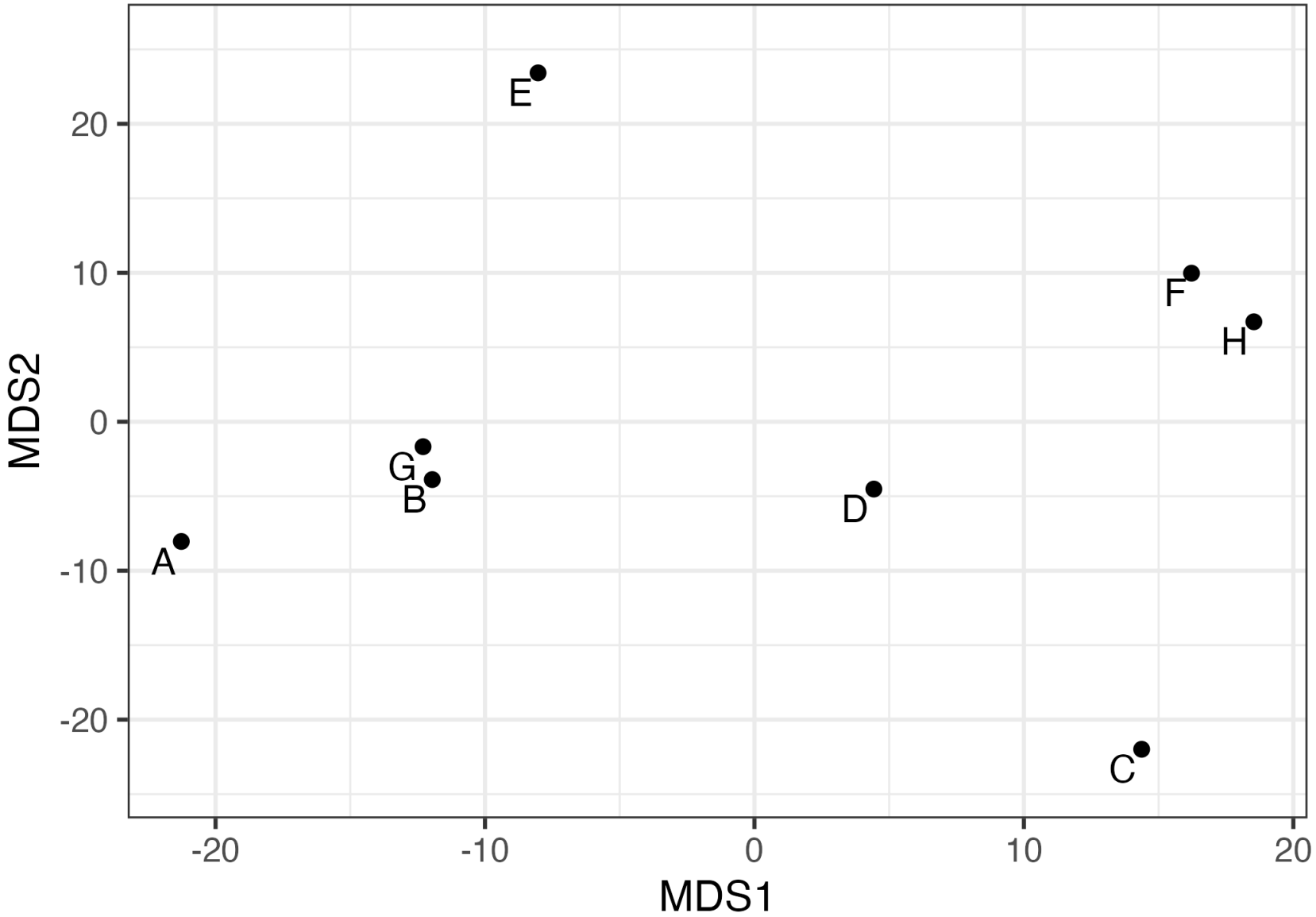
Метод Варда

= Ward's Minimum Variance Clustering
объекты объединяются в кластеры так, чтобы внутригрупповая дисперсия расстояний была минимальной.



Особенности
метод годится и для неевклидовых расстояний несмотря на то, что внутригрупповая дисперсия расстояний рассчитывается так, как будто это евклидовы расстояния.

Как работает метод Варда



Кластерный анализ в R

Кластеризация

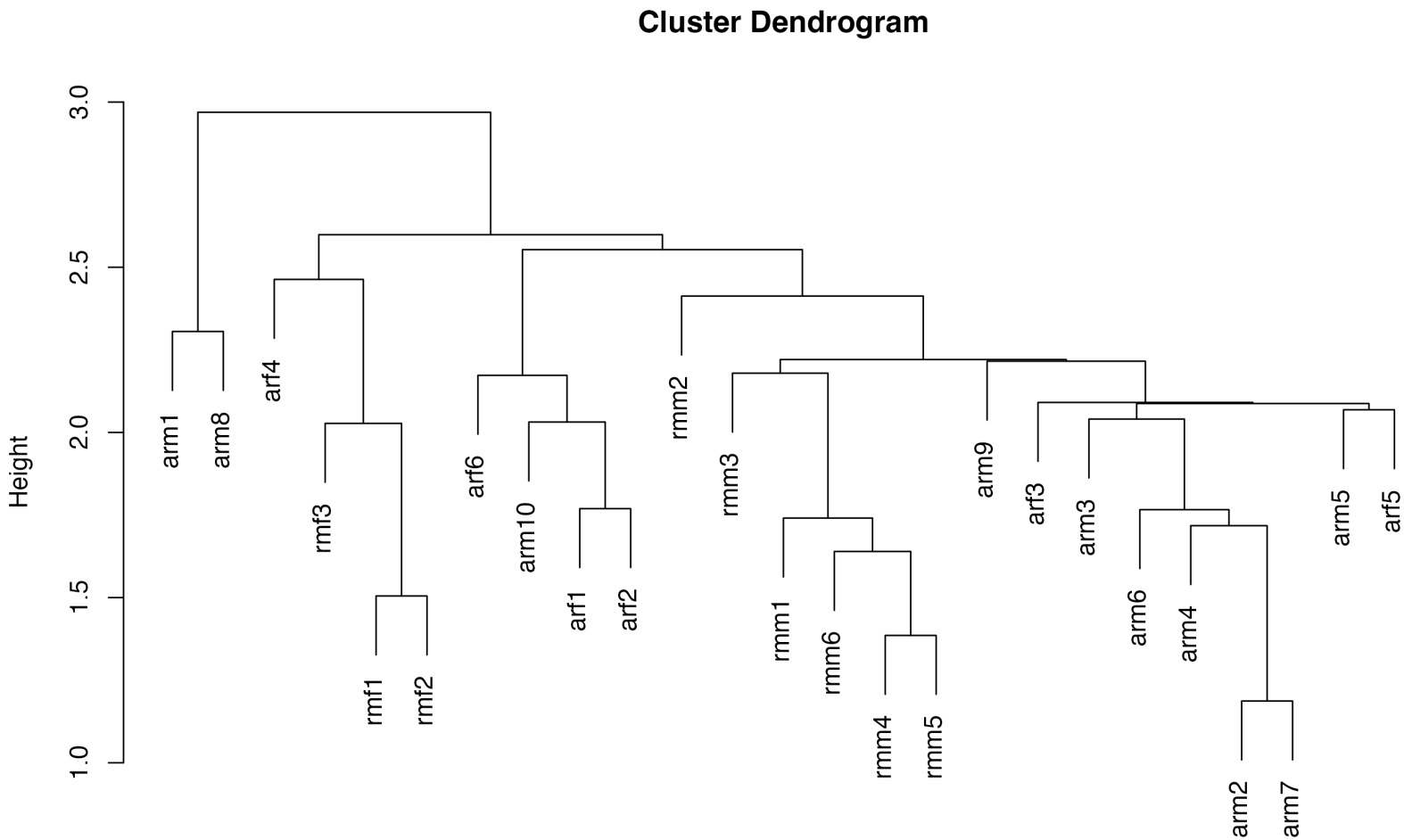
Давайте построим деревья при помощи нескольких алгоритмов кластеризации (по стандартизованным данным, с использованием Евклидова расстояния) и сравним их.

```
# Пакеты для визуализации кластеризации
library(ape)
library(dendextend)
```

```
# Матрица расстояний
d <- dist(x = st_w, method = "euclidean")
```

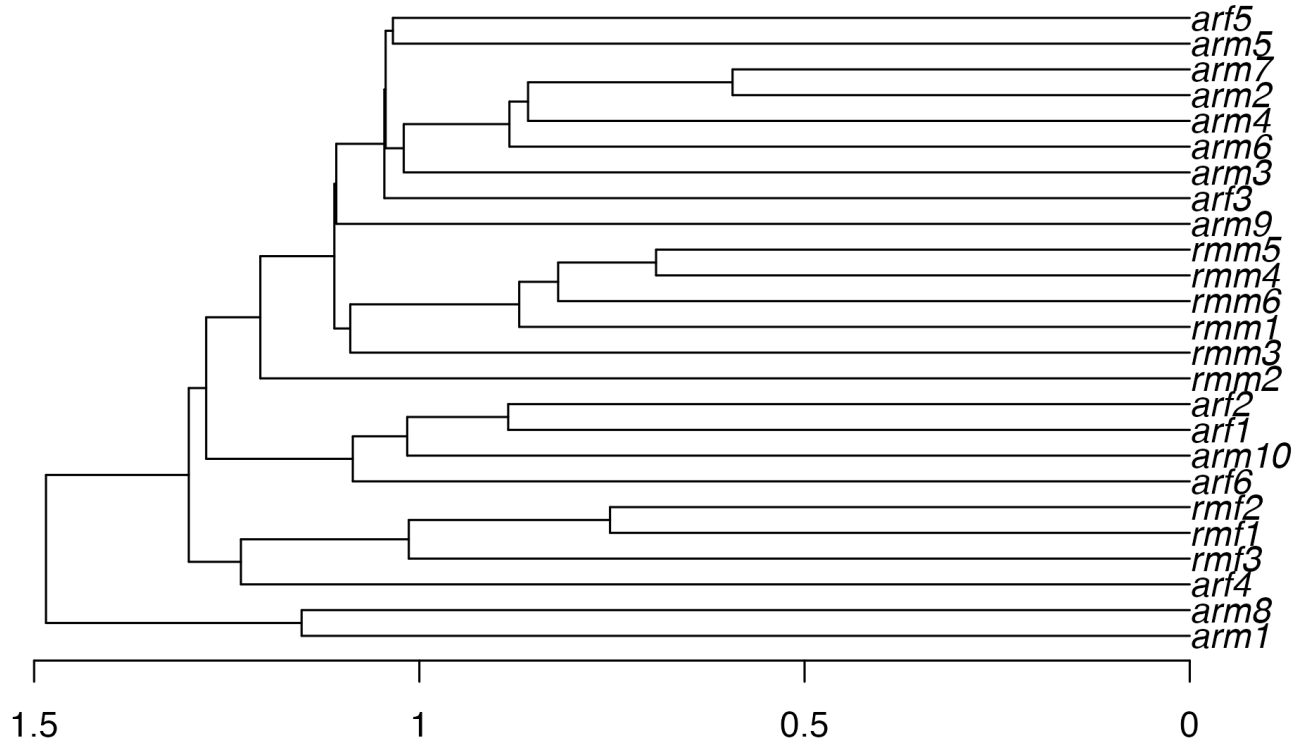
(1.0) Метод ближайшего соседа + base

```
hc_single <- hclust(d, method = "single")
plot(hc_single)
```



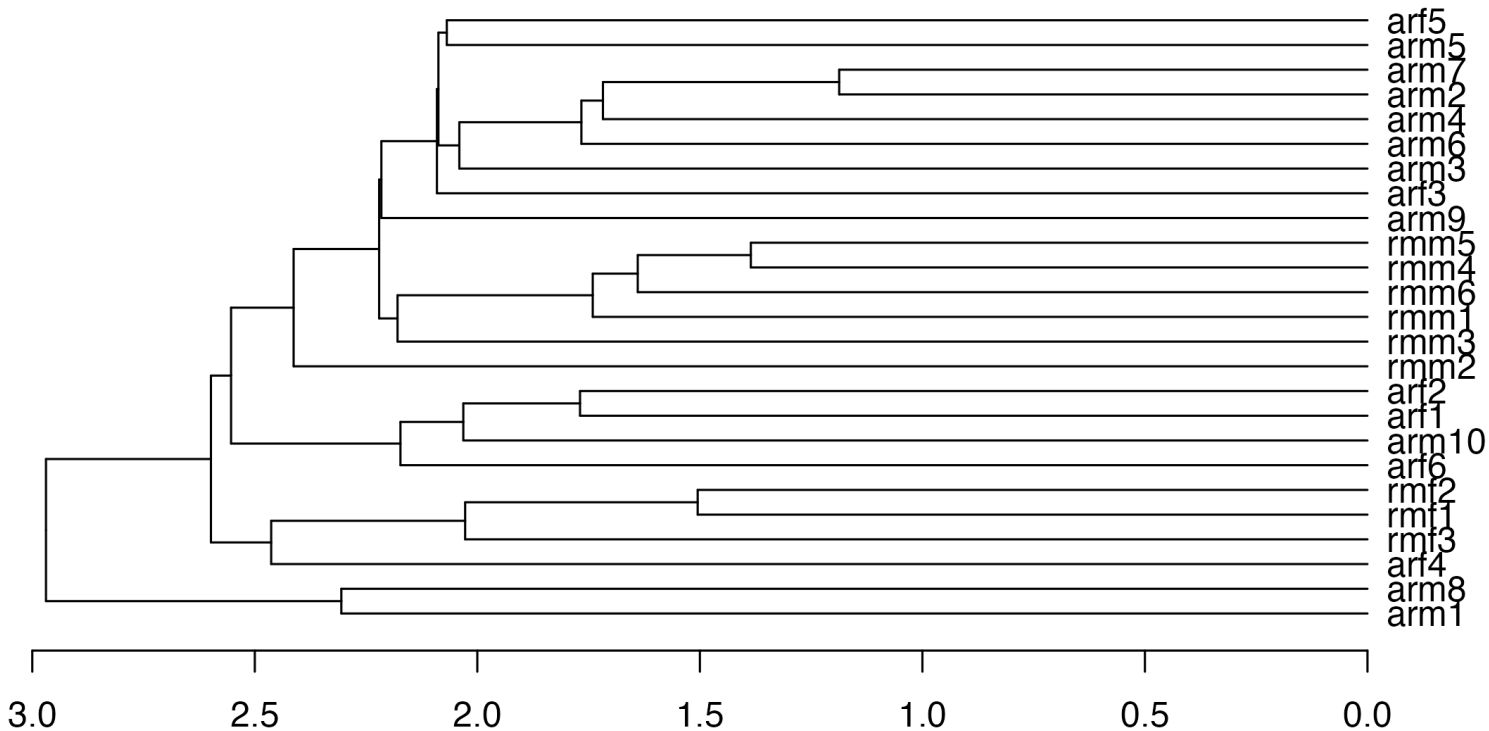
(1.1) Метод ближайшего соседа + ape

```
ph_single <- as.phylo(hc_single)
plot(ph_single, type = "phylogram")
axisPhylo()
```



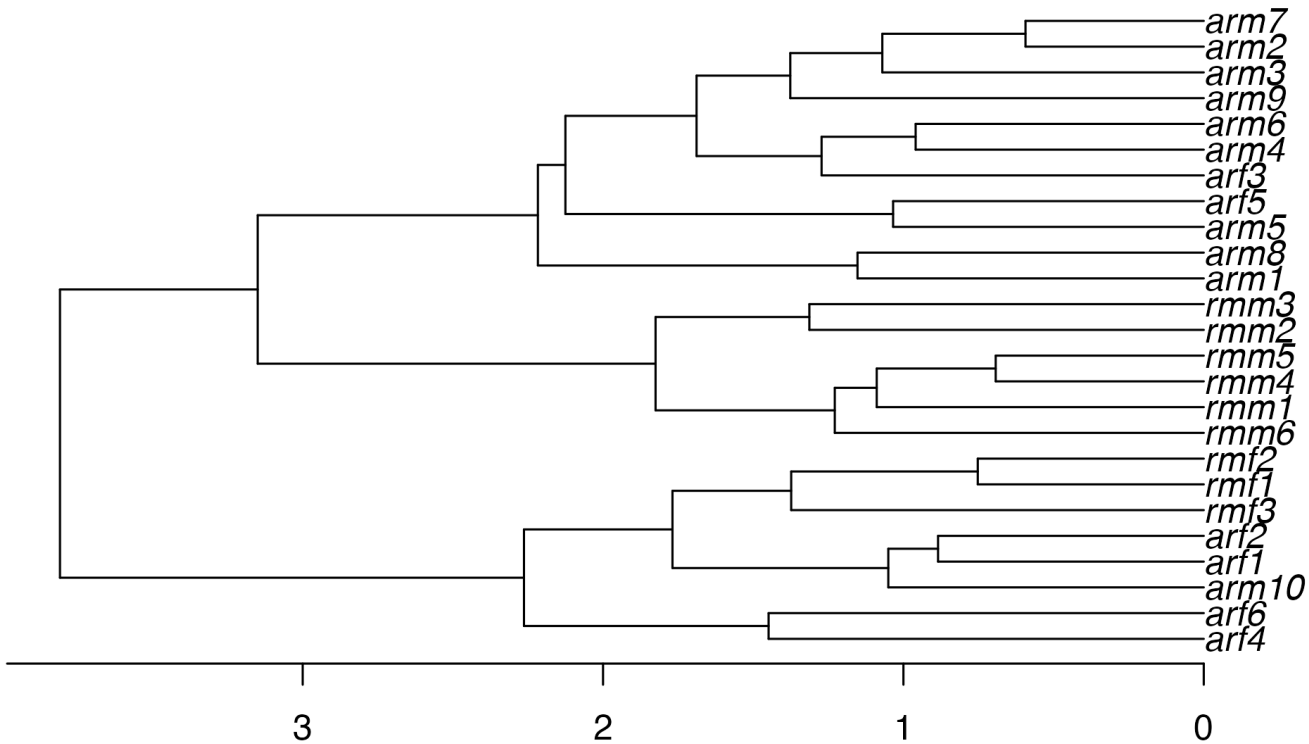
(1.2) Метод ближайшего соседа + dendextend

```
den_single <- as.dendrogram(hc_single)
plot(den_single, horiz = TRUE)
```



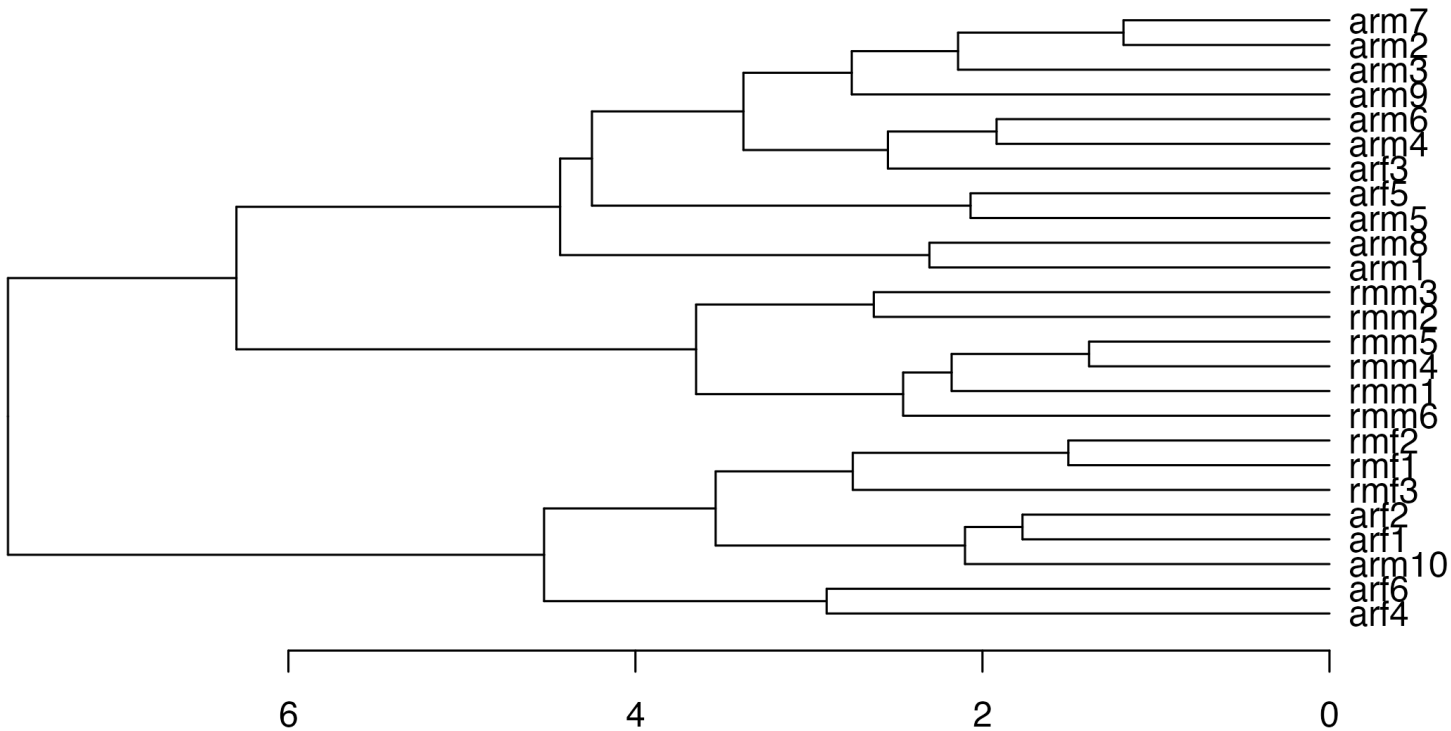
(2.1) Метод отдаленного соседа + ape

```
hc_compl <- hclust(d, method = "complete")
ph_compl <- as.phylo(hc_compl)
plot(ph_compl, type = "phylogram")
axisPhylo()
```



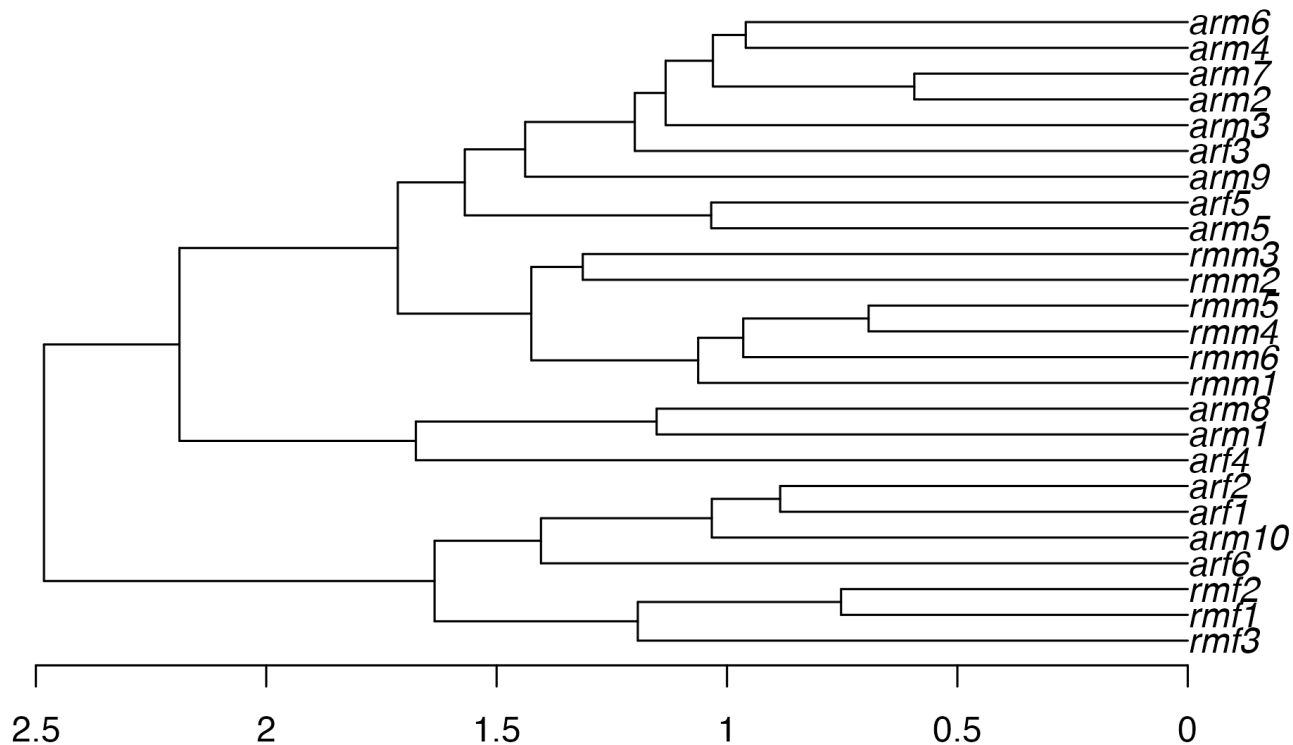
(2.2)Метод отдаленного соседа + dendextend

```
den_compl <- as.dendrogram(hc_compl)
plot(den_compl, horiz = TRUE)
```



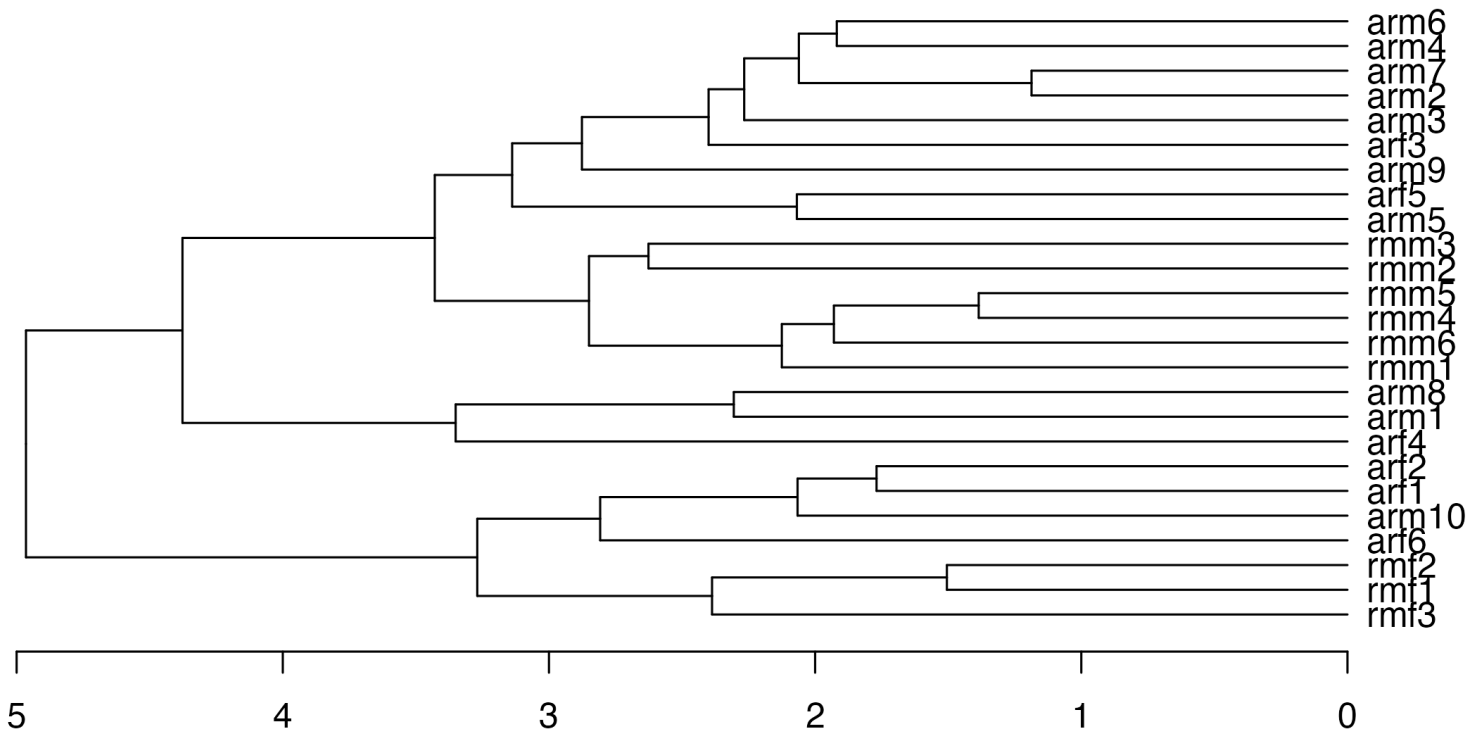
(3.1) Метод невзвешенного попарного среднего (UPGMA) + ape

```
hc_avg <- hclust(d, method = "average")
ph_avg <- as.phylo(hc_avg)
plot(ph_avg, type = "phylogram")
axisPhylo()
```



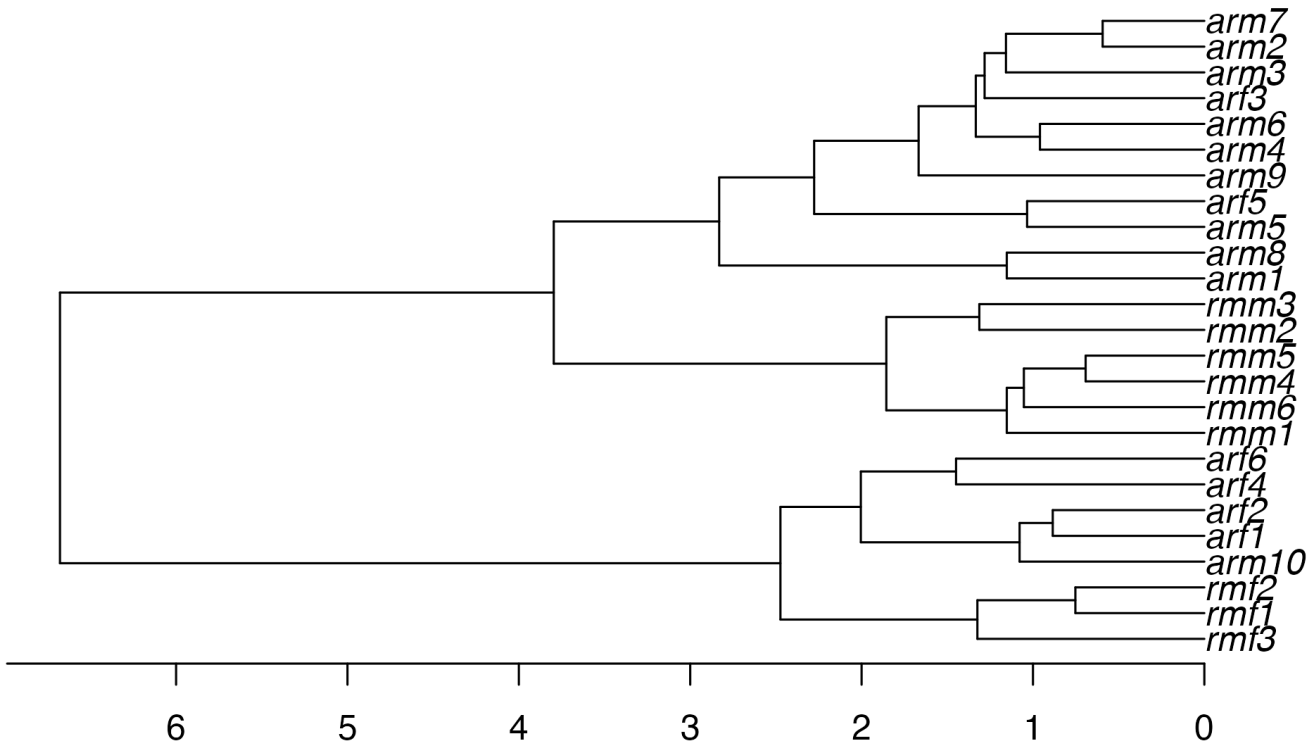
(3.2) Метод невзвешенного попарного среднего (UPGMA) + dendextend

```
den_avg <- as.dendrogram(hc_avg)
plot(den_avg, horiz = TRUE)
```



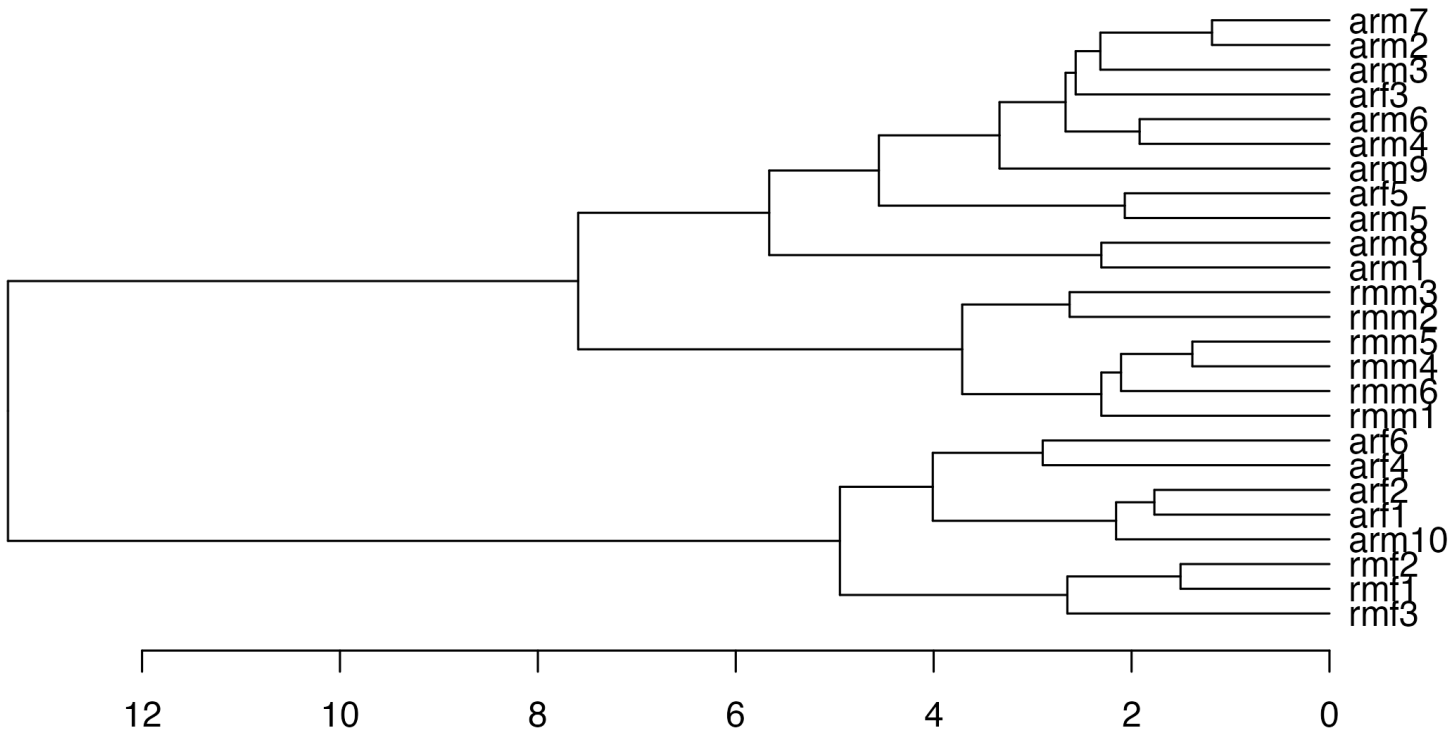
(4.1) Метод Варда + ape

```
hc_w2 <-hclust(d, method = "ward.D2")
ph_w2 <- as.phylo(hc_w2)
plot(ph_w2, type = "phylogram")
axisPhylo()
```



(4.2) Метод Варда + dendextend

```
den_w2 <- as.dendrogram(hc_w2)
plot(den_w2, horiz = TRUE)
```



Качество кластеризации

Кофенетическая корреляция: расчёт

Кофенетическое расстояние — расстояние между объектами на дендрограмме, т.е. то расстояние, на котором объекты становятся частью одной группы в ходе процесса кластеризации.

Кофенетическая корреляция — мера качества отображения многомерных данных на дендрограмме. Кофенетическую корреляцию можно рассчитать как Пирсоновскую корреляцию (обычную) между матрицами исходных и кофенетических расстояний между всеми парами объектов. В идеальном случае равна 1.

$$r = \frac{\sum_{i<j}(d_{ij} - \bar{d})(c_{ij} - \bar{c})}{\sqrt{\sum_{i<j}(d_{ij} - \bar{d})^2 \cdot \sum_{i<j}(c_{ij} - \bar{c})^2}}$$

где:

- d_{ij} — исходное расстояние между объектами i и j,
- c_{ij} — кофенетическое расстояние между объектами i и j,
- $\bar{d}$ — среднее исходных расстояний,
- $\bar{c}$ — среднее кофенетических расстояний.

Кофенетическая корреляция: смысл

Метод агрегации, который дает наибольшую кофенетическую корреляцию, дает кластеры, лучше всего отражающие исходные данные.

Матрицу кофенетических расстояний и кофенетическую корреляцию можно рассчитать при помощи функций из пакета `stats` (`dendextend`) и ape, соответственно.

Кофенетическая корреляция в R

```
# Матрица кофенетических расстояний
c_single <- cophenetic(ph_single)

# Кофенетическая корреляция =
# = корреляция матриц кофенетич. и реальн. расстояний
cor(d, as.dist(c_single))
```

```
[1] 0.5654072
```

Задание:

Оцените при помощи кофенетической корреляции качество кластеризаций, полученных разными методами.

Какой метод дает лучший результат?

Решение

```
c_single <- cophenetic(ph_single)
cor(d, as.dist(c_single))
```

```
[1] 0.5654072
```

```
c_compl <- cophenetic(ph_compl)
cor(d, as.dist(c_compl))
```

```
[1] 0.705757
```

```
c_avg <- cophenetic(ph_avg)
cor(d, as.dist(c_avg))
```

```
[1] 0.7446591
```

```
c_w2 <- cophenetic(ph_w2)
cor(d, as.dist(c_w2))
```

```
[1] 0.7259618
```

Что можно делать дальше с дендрограммой

- Можно выбрать число кластеров:
  - либо субъективно, на любом выбранном уровне (главное, чтобы кластеры были осмысленными и интерпретируемыми);
  - либо исходя из распределения расстояний ветвления.
- Можно оценить стабильность кластеризации при помощи бутстрепа.

Ширина силуэта

Ширина силуэта s_i — мера степени принадлежности объекта i к кластеру

$$s_i = \frac{\bar{d}_{i\ to\ nearest\ cluster} - \bar{d}_{i\ within}}{max\{\bar{d}_{i\ to\ nearest\ cluster}, \bar{d}_{i\ within}\}}$$

s_i — сравнивает между собой средние расстояния от данного объекта:

- $\bar{d}_{i\ within}$ — до других объектов из того же кластера
- $\bar{d}_{i\ to\ nearest\ cluster}$ — до ближайшего кластера

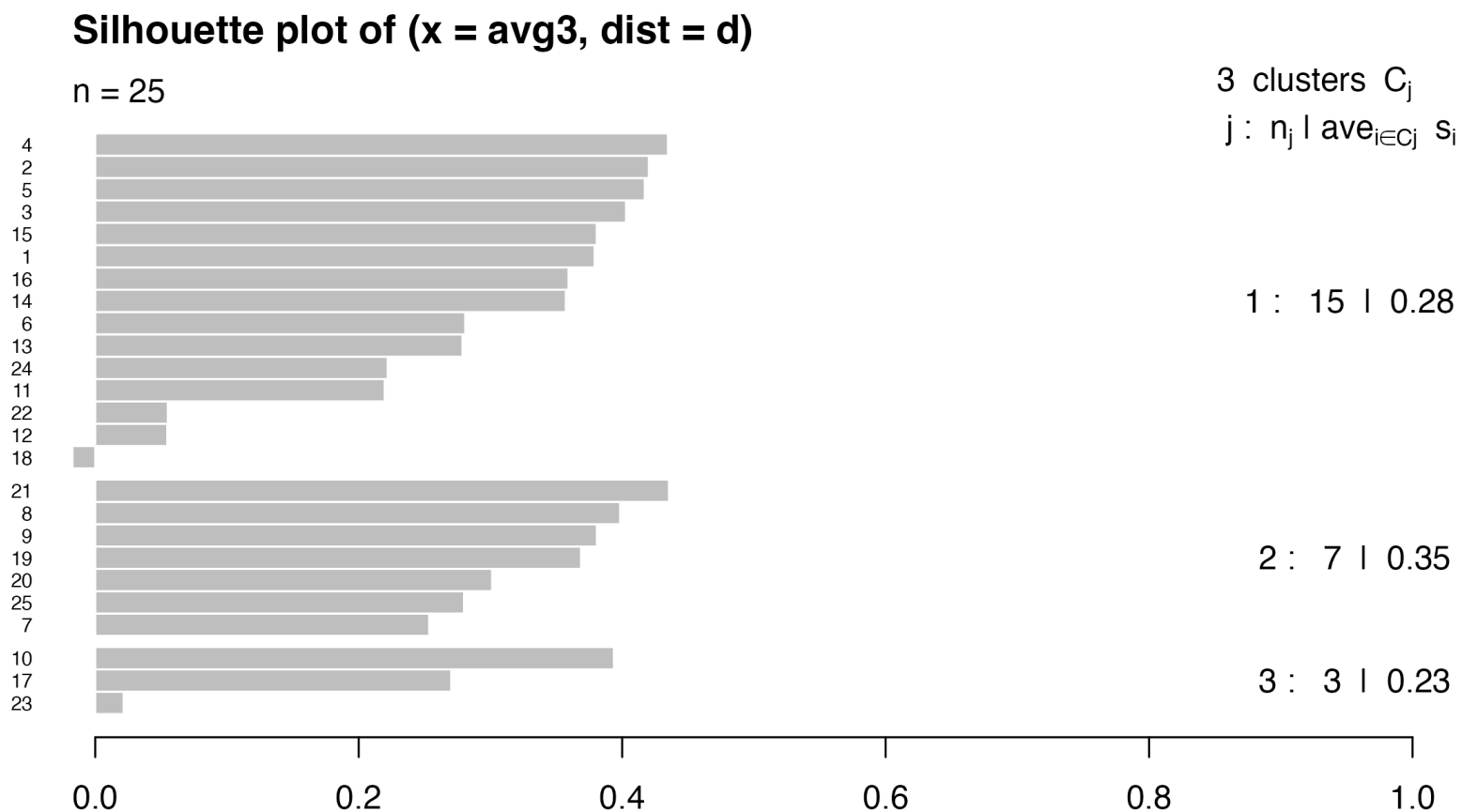
$-1 \leq s_i \leq 1$ — чем больше s_i, тем "лучше" объект принадлежит кластеру.

- Средняя ширина силуэта для всех объектов из кластера — оценивает, насколько "тесно" сгруппированы объекты.
- Средняя ширина силуэта по всем данным — оценивает общее качество классификации.
- Чем больше к 1, тем лучше. Если меньше 0.25, то можно сказать, что нет структуры.

Как рассчитывается ширина силуэта

Оценим ширину силуэта для 3 кластеров

```
library(cluster)
avg3 <- cutree(tree = hc_avg, k = 3) # делим дерево на нужное количество
↪  кластеров
plot(silhouette(x = avg3, dist = d), cex.names = 0.6)
```



Бутстреп

Бутстреп — один из методов оценки значимости полученных результатов; повторная выборка из имеющихся данных. В такой выборке элементы могут повторяться.

Алгоритм

- Имеющиеся данные делим на группы одинакового размера: какие-то элементы могут отсутствовать, а какие-то — повторяться несколько раз.
- На основе полученных данных строим дендрограмму.
- Повторяем всё много раз.

Бутстреп поддержка ветвей

> *"An approximately unbiased test of phylogenetic tree selection."*
>
> *— Shimodaria, 2002*

Этот тест использует специальный вариант бутстрепа — multiscale bootstrap. Мы не просто многократно берем бутстреп-выборки и оцениваем для них вероятность получения топологий (BP p-value), эти выборки еще и будут с разным числом объектов. По изменению BP при разных объемах выборки можно вычислить AU (approximately unbiased p-value). BP недооценивает поддержку истинной топологии из-за дискретности повторных выборок.

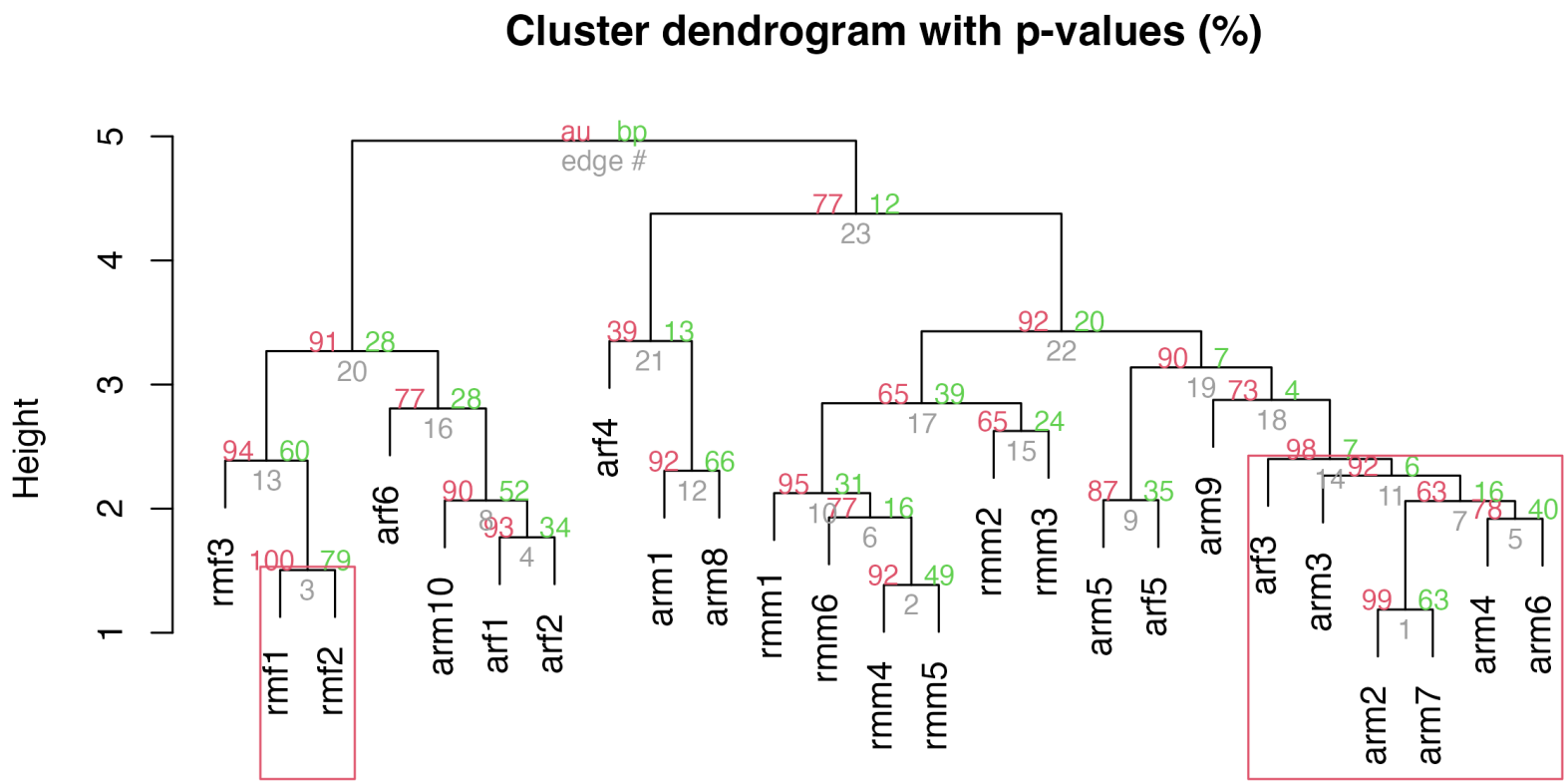
```
library(pvclust)
set.seed(389)
# итераций должно быть 1000 и больше, здесь мало для скорости
cl_boot <- pvclust(t(st_w), method.hclust = "average", nboot = 100,
                   method.dist = "euclidean", parallel = TRUE, iseed = 42)
```

Обратите внимание на число итераций: `nboot = 100` — это очень мало. На самом деле нужно 10000 или больше.

Дерево с величинами поддержки

AU — approximately unbiased p-values (красный), BP — bootstrap p-values (зеленый).

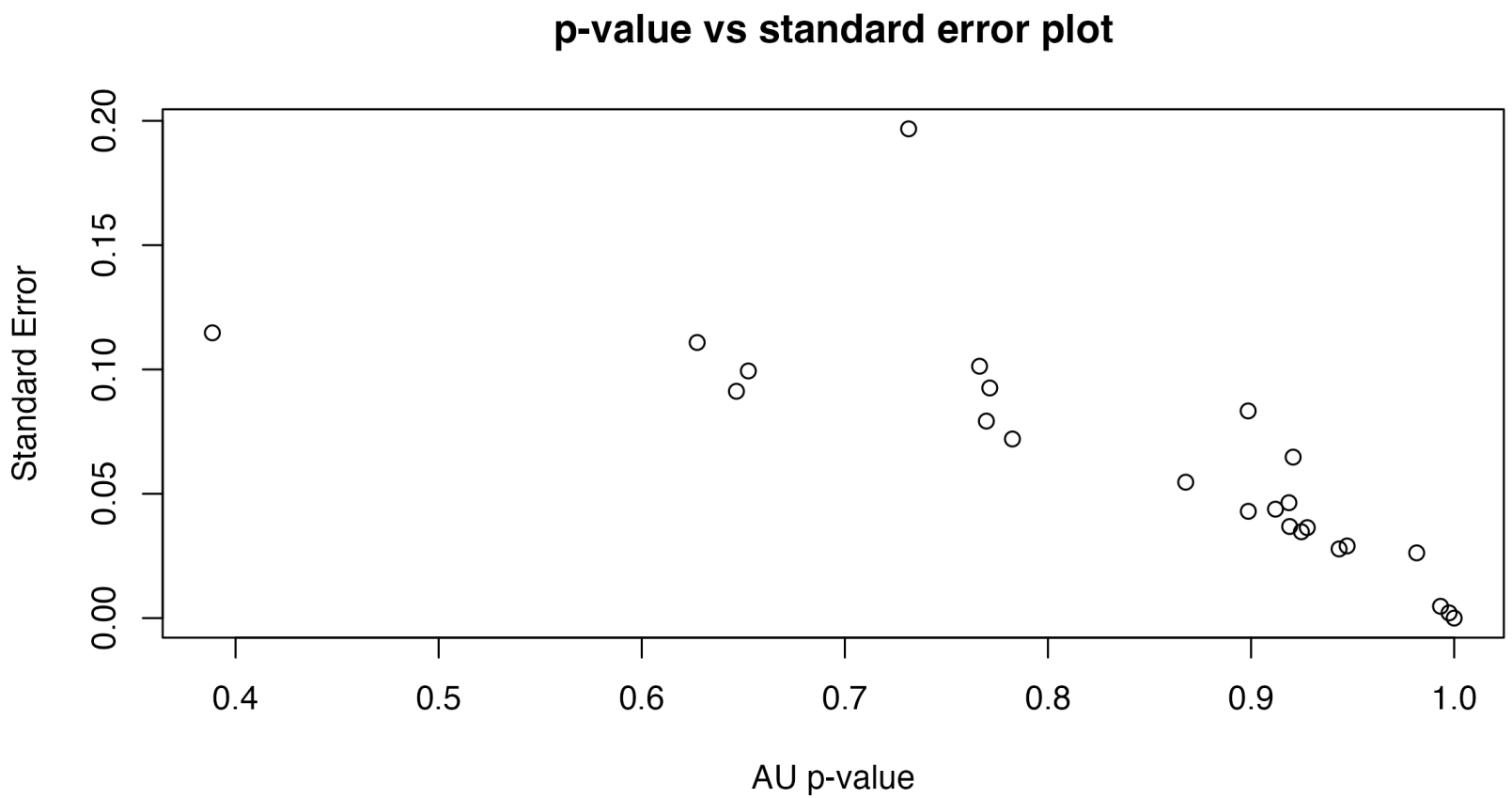
```
plot(cl_boot)
pvrect(cl_boot) # достоверные ветвления
```



Для диагностики качества оценок AU

График стандартных ошибок для AU p-value нужен, чтобы оценить точность оценки самих AU. Чем больше было бутстреп-итераций, тем точнее будет оценка AU.

```
seplot(cl_boot)
# print(cl_boot) # все значения
```



Сопоставление деревьев: Танглграммы

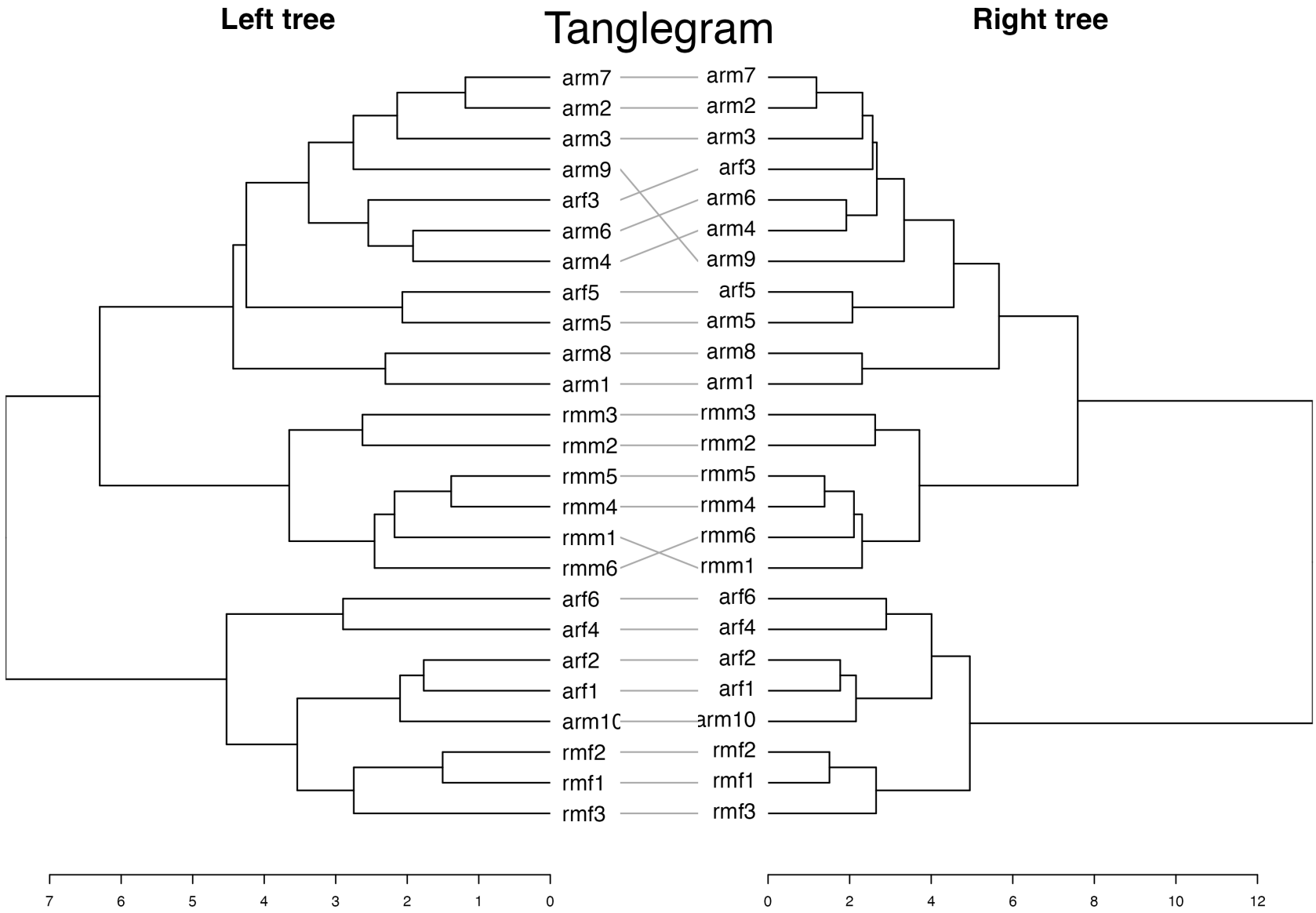
Танглграмма

Два дерева (с непохожим ветвлением) выравнивают, вращая случайным образом ветви вокруг оснований. Итеративный алгоритм. Картина каждый раз разная.

```
set.seed(395)
untang_w <- untangle_step_rotate_2side(den_compl, den_w2, print_times = F)

# танглграмма
tanglegram(untang_w[[1]], untang_w[[2]],
           highlight_distinct_edges = FALSE,
           common_subtrees_color_lines = F,
           main = "Tanglegram",
           main_left = "Left tree",
           main_right = "Right tree",
           columns_width = c(8, 1, 8),
           margin_top = 3.2, margin_bottom = 2.5,
           margin_inner = 4, margin_outer = 0.5,
           lwd = 1.2, edge.lwd = 1.2,
           lab.cex = 1.5, cex_main = 2)
```

Танглграмма



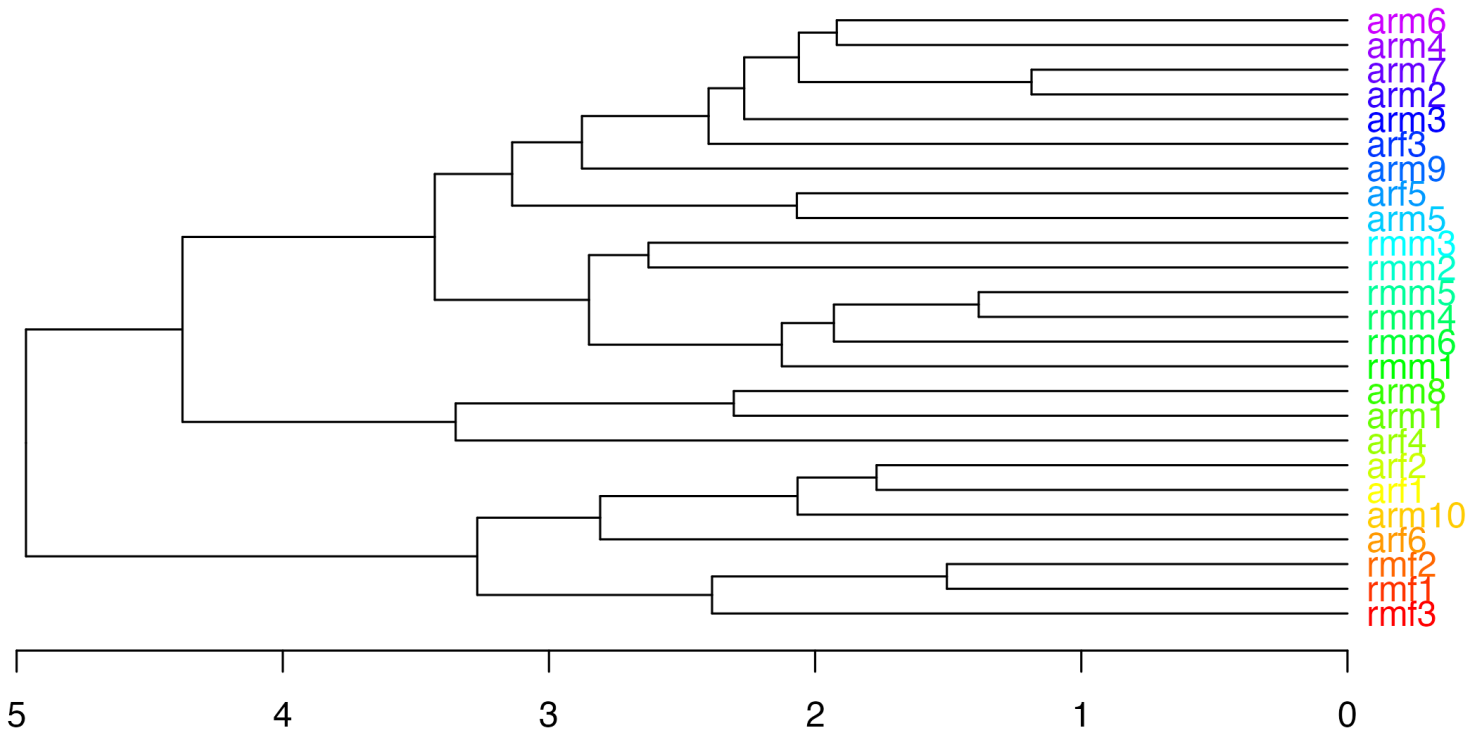
Задание

Постройте танглграмму из дендрограмм, полученных методом ближайшего соседа и методом Варда.

Раскраска дендрограмм

Вручную

```
# Произвольные цвета радуги
cols <- rainbow(30)
den_avg_manual <- color_labels(dend = den_avg, col = cols)
plot(den_avg_manual, horiz = TRUE)
```

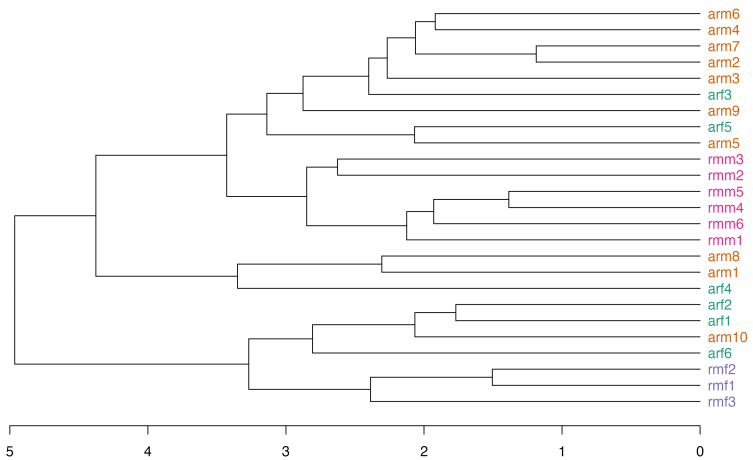


Раскраска дендрограмм

С помощью функции

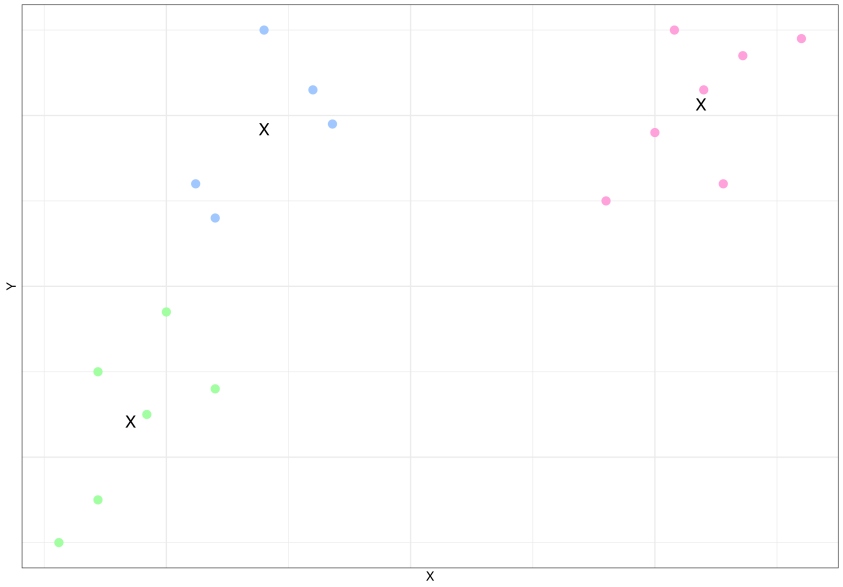
```
# Функция для превращения лейблов в
↪  цвета
# (группы определяются по `n_chars`
↪  первых букв в лейбле)
get_colours <- function(dend,
↪  n_chars, palette = "Dark2"){
  labs <- get_leaves_attr(dend,
↪  "label")
  group <- substr(labs, start = 0,
↪  stop = n_chars)
  group <- factor(group)
  cols <-
↪  brewer.pal(length(levels(group)),
↪  name = palette)[group]
  return(cols)
}
```

```
library(RColorBrewer)
cols <- get_colours(dend = den_avg,
↪  n_chars = 3)
den_avg_c <- color_labels(dend =
↪  den_avg, col = cols)
plot(den_avg_c, horiz = TRUE)
```



Неиерархические методы кластеризации

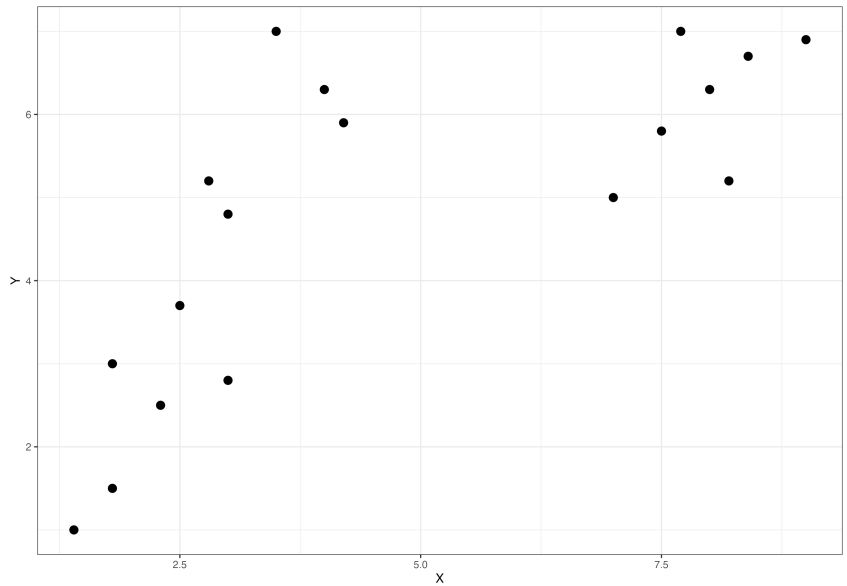
Метод K-средних (K-means)



В отличие от иерархических методов кластеризации K-means будет искать то количество кластеров (k), которое вы ему зададите. Каждое наблюдение принадлежит кластеру с ближайшим значением среднего числа (центроида); помимо этого K-means кластеризация минимизирует разброс значений внутри каждого из кластера. Используется в машинном обучении, в том числе, например, для цветовой редуцкии изображений.

Алгоритм K-means

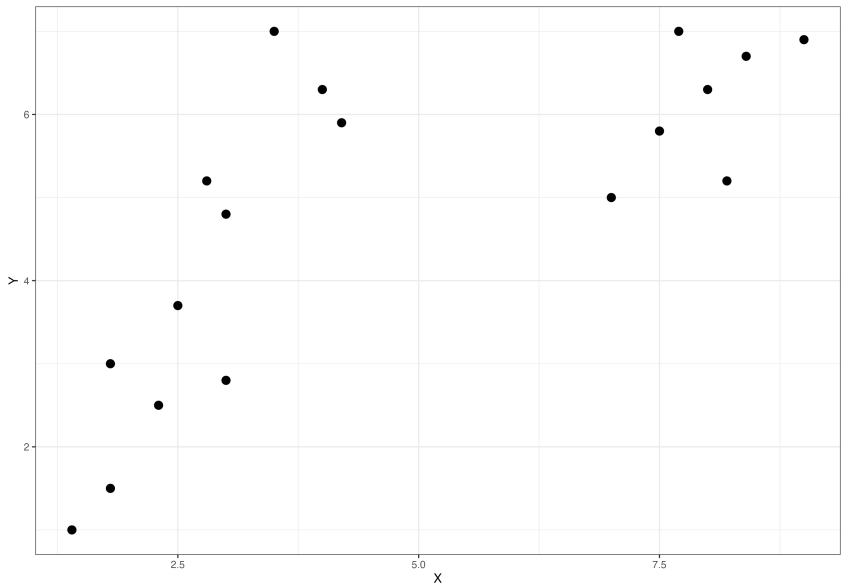
График наблюдений



Здесь как будто бы выделяются 3 кластера, поэтому возьмём k = 3.
Что же будет делать алгоритм?

Алгоритм K-means

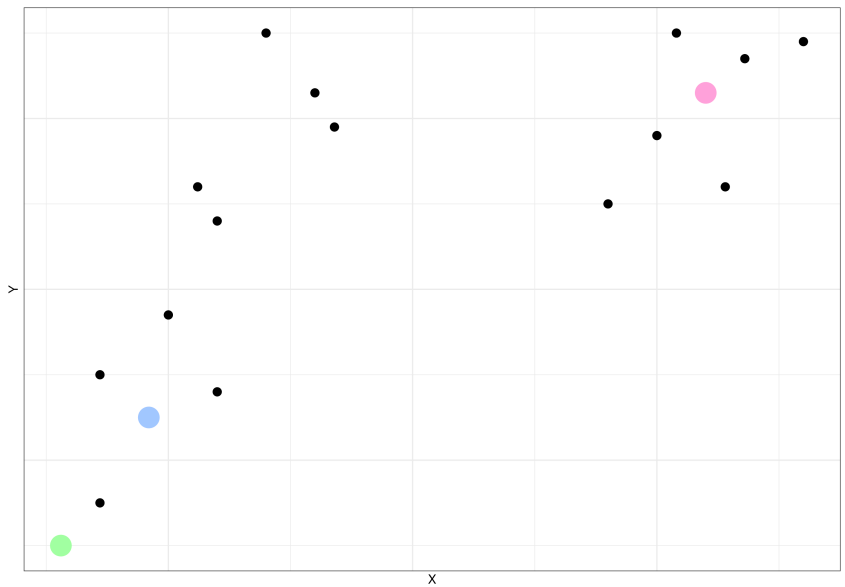
График наблюдений



Здесь как будто бы выделяются 3 кластера, поэтому возьмём k = 3.
Что же будет делать алгоритм?

1. Выбираются случайным образом 3 точки на графике — кластерные центроиды

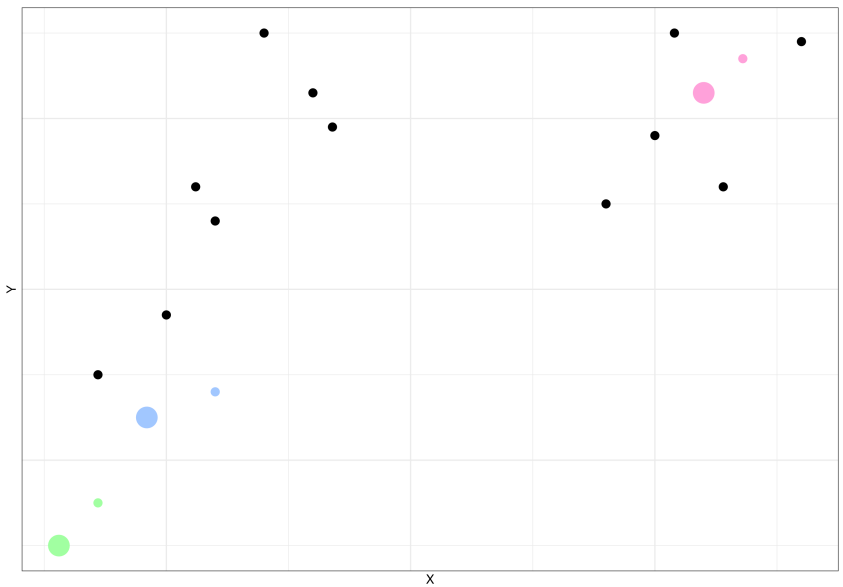
Например, так:



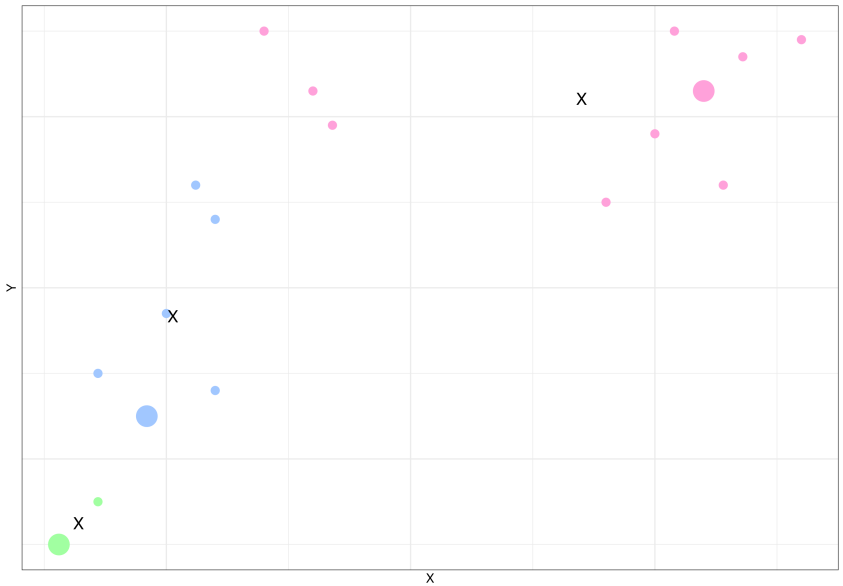
Алгоритм K-means

2. Измеряется Евклидово расстояние между каждой точкой и центроидом

При этом каждая точка приписывается к ближайшему кластеру.

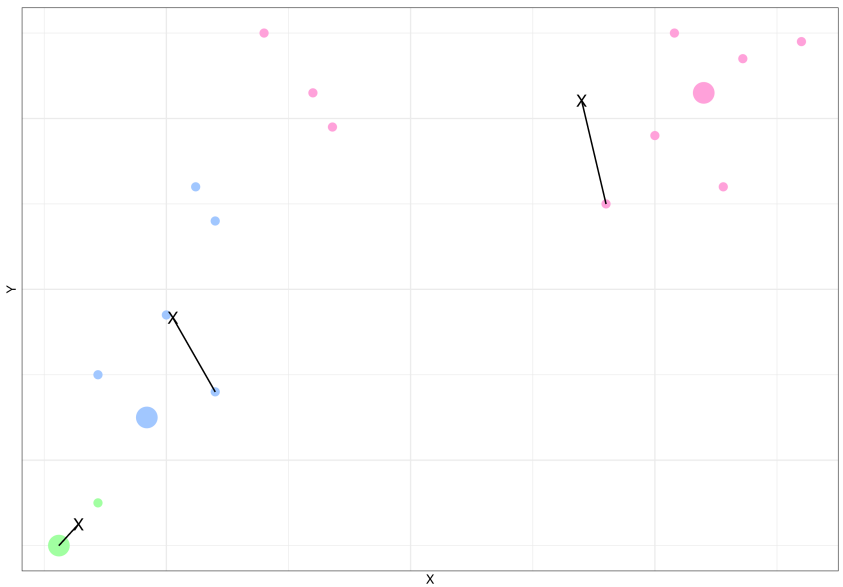


3. Расситываются центроиды для каждого кластера



4. Расчёт расстояний от каждой точки до нового центроида

Также оценивается разброс внутри каждого кластера.

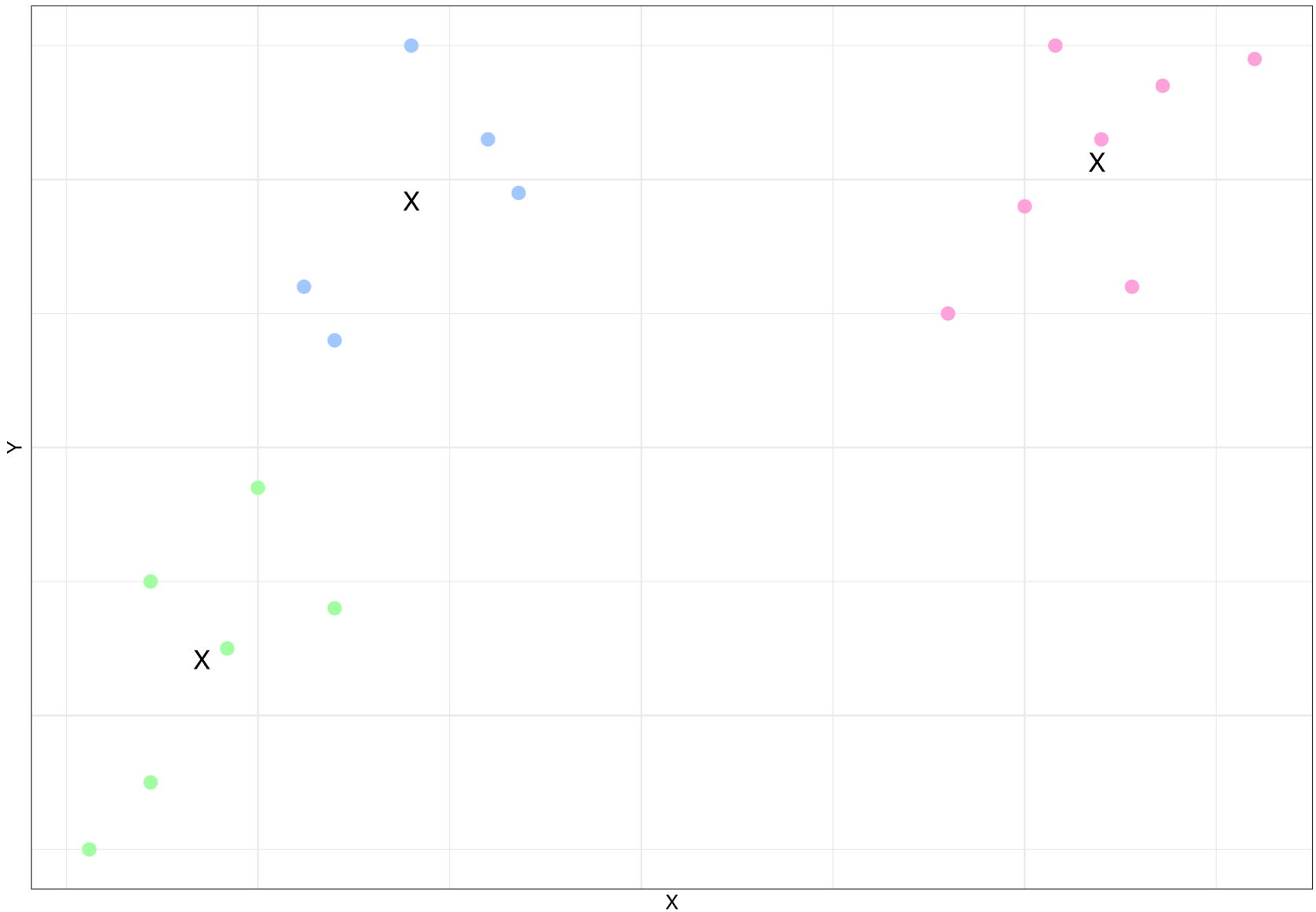


Разброс считается как сумма квадратов расстояний между отдельными наблюдениями и центроидом.

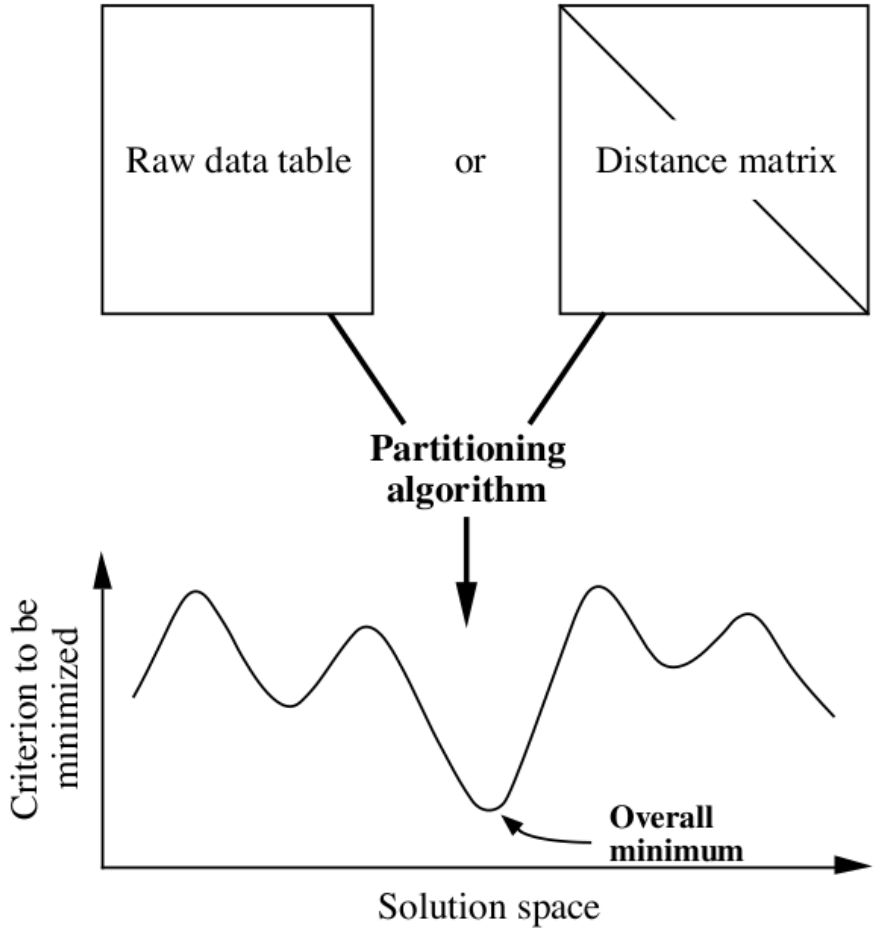
$$\sum_{i=1}^{n}(x_i - \overline{x})$$

5. Повторяем всё многократно до тех пор, пока разброс не станет минимальным

Кластеры с минимальным разбросом — финальные.



Поиск локального минимума



Из Legendre, Legendre, 2012

Кратко алгоритм можно свести к поиску общего минимума среди локальных (вне зависимости от начальной конфигурации).

K-means в R

Если в данных много нулей, их обязательно нужно стандартизовать (что мы уже делали).

K-means кластеризацию в R делает функция `kmeans` (пакет `stats`), ей нужно передать аргументы `centers` (количество кластеров) и `nstart` (количество случайных итераций).

```
set.seed(333)
w_kmeans <- kmeans(st_w, centers = 3, nstart = 100)
```

Полученные результаты можно сравнить с тем, что нам дала иерархическая кластеризация (UPGMA) — оценка ширины силуэта.

```
table(avg3, w_kmeans$cluster)
```

```
avg3  1  2  3
   1  6  0  9
   2  0  7  0
   3  0  1  2
```

Как выбрать нужное количество кластеров?

Два популярных критерия:

- Индекс Калински-Харабаза (Calinski–Harabasz index): F-статистика, сравнивающая меж- и внутригрупповую сумму квадратов. Если группы одинаковой величины.
- Индекс простой структуры (Simple Structure Index): оценивает влияние на интепретабельность полученной кластеризации. Если группы разной величины.

Критерии выбора количества кластеров

Индекс Калински-Харабаза

$$CH(k) = \frac{SS_B/(k-1)}{SS_W/(n-k)}$$

где:

- SS_B — сумма квадратов расстояний от центроида кластера до общего центроида, умноженное на количество объектов в кластере,
- SS_W — сумма квадратов расстояний до центроида собственного кластера,
- k — число кластеров,
- n — количество исходных наблюдений.

Выбирается число кластеров с **наибольшим** значением CH.

Индекс простой структуры

$$SSI = \frac{1}{n}\sum_{i=1}^{n}\frac{b_i - a_i}{\max(a_i, b_i)}$$

где:

- a_i — расстояние от объекта i до центроида своего кластера,
- b_i — расстояние от объекта i до ближайшего центроида другого кластера,
- n — количество исходных наблюдений по всем кластерам.

Выбирается число кластеров с **наибольшим** значением SSI.

Как выбрать нужное количество кластеров в R

Функция `cascadeKM` из пакета `vegan`. По сути функция-обёртка, которая проводит кластеризацию с разным заданным количеством кластеров.

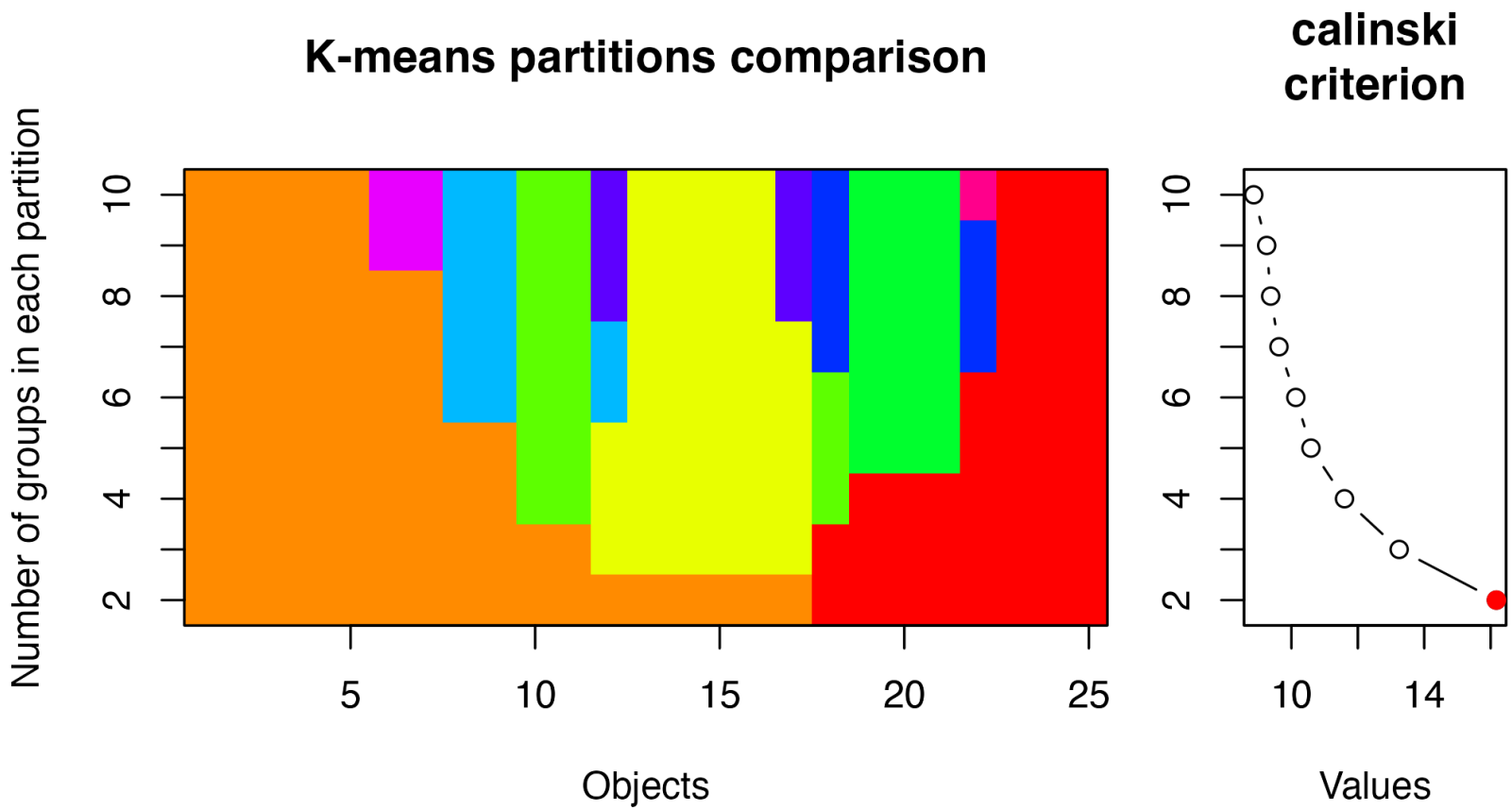
```
w_cascade <- cascadeKM(st_w, inf.gr = 2, sup.gr = 10,
                       iter = 100, criterion = 'calinski')
```

- `inf.gr` — начальное количество кластеров,
- `sup.gr` — максимальное количество кластеров,
- `iter` — количество итераций для каждой кластеризации,
- `criterion` — индекс: `calinski` или `ssi`.

Визуализация результатов множественной кластеризации

Рисуем так, чтобы объекты, относящиеся к одному кластеру, рисовались вместе.

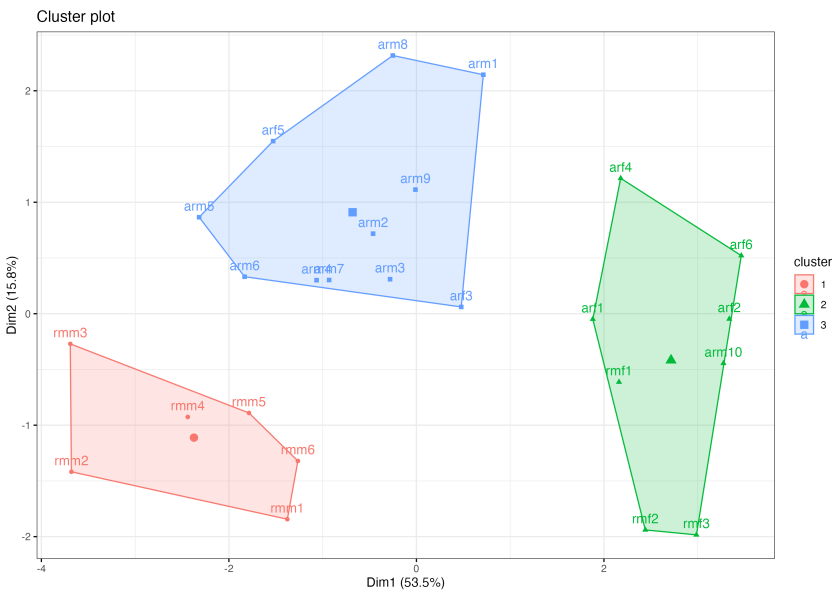
```
plot(w_cascade, sortg = TRUE)
```



Визуализация k-means

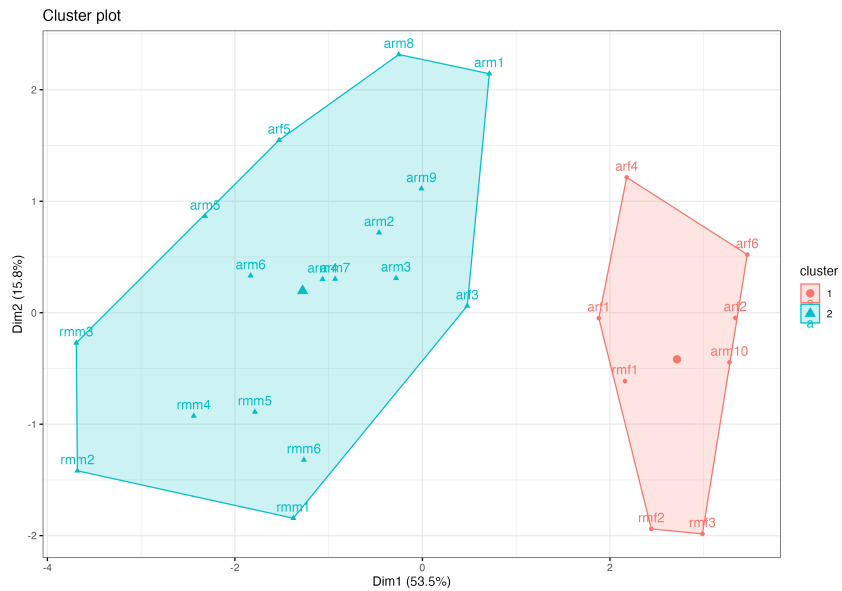
Делается с помощью функции `fviz_cluster` из пакета `factoextra`.

```
library(factoextra)
fviz_cluster(w_kmeans, data = st_w,
             ggtheme = theme_bw())
```



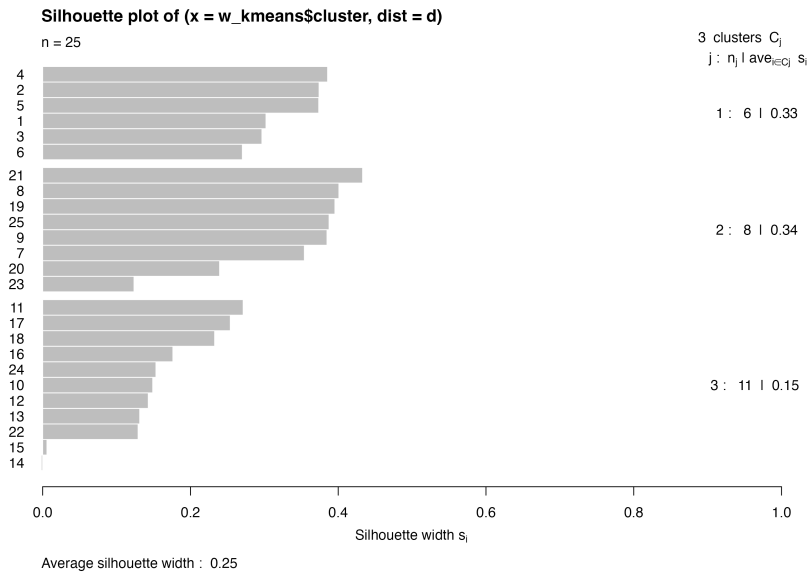

```
w_2k <- kmeans(st_w, centers = 2,
               nstart = 100)

fviz_cluster(w_2k, data = st_w,
             ggtheme = theme_bw())
```

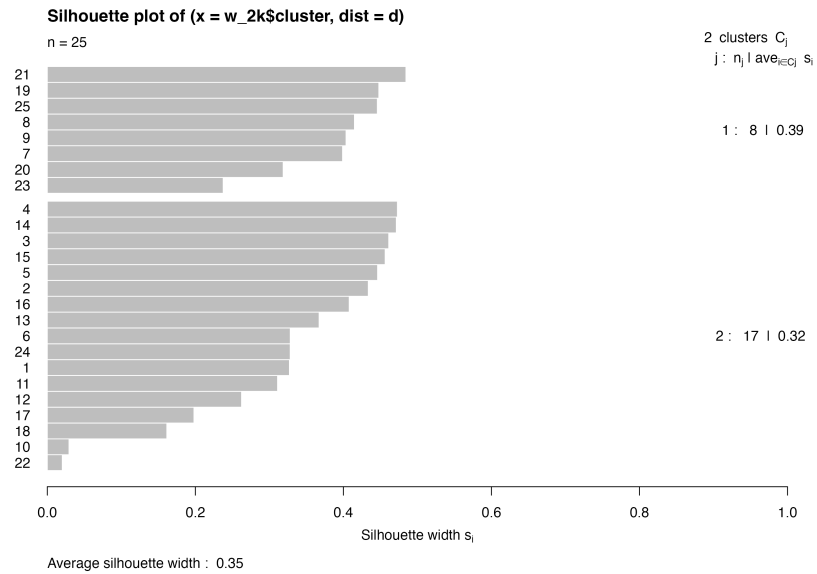


Оценка качества кластеризации: ширина силуэта

```
plot(silhouette(w_kmeans$cluster, d))
```

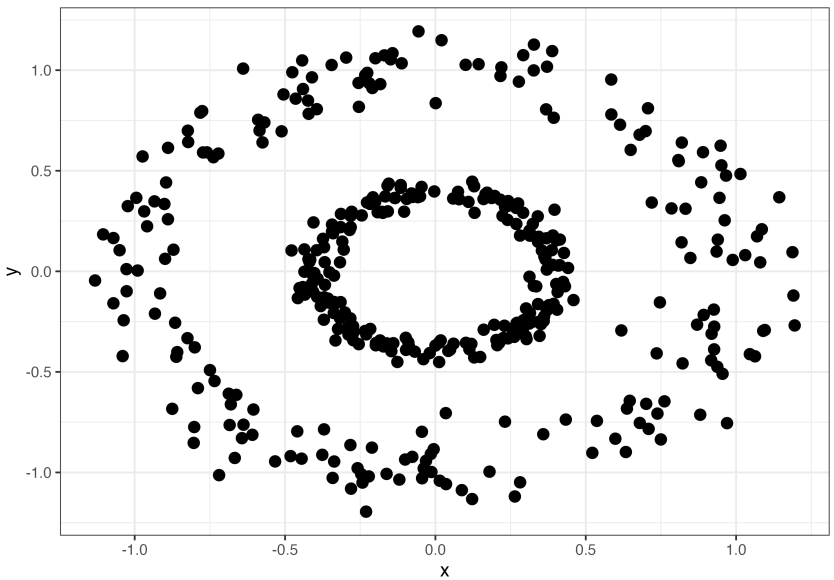



```
plot(silhouette(w_2k$cluster, d))
```



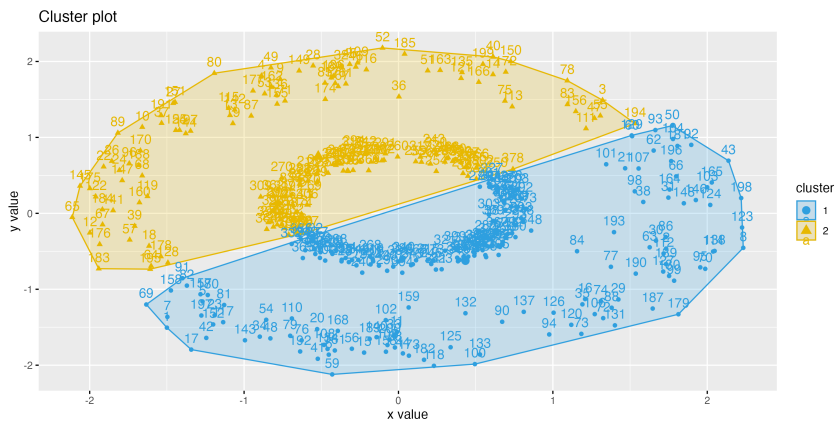
Density-based spatial clustering of applications with noise (DBSCAN)

Основанная на плотности пространственная кластеризация для приложений с шумами — метод, более подходящий для “вложенных” кластеров. Основан на распределении плотности точек.



Работает с данными, с которыми другие методы кластеризации не могут справиться (K-means в примере).

```
set.seed(123)
circle_kmeans <- kmeans(multi,
↪  centers = 2, nstart = 20)
my_col_circle <- c("#2E9FDF",
↪  "#E7B800")
fviz_cluster(circle_kmeans,
             data = multi, palette =
             ↪  my_col_circle)
```



Принцип работы DBSCAN

Кластеры выбираются на основе плотности расположения точек. В результате в единый кластер объединяются близко расположенные друг к другу точки.

Задаваемые параметры:

- радиус расстояния, на котором должны рассматриваться близлежащие точки (`eps`)
- минимальное количество точек, которые расположены в круге этого радиуса (`minPts`)

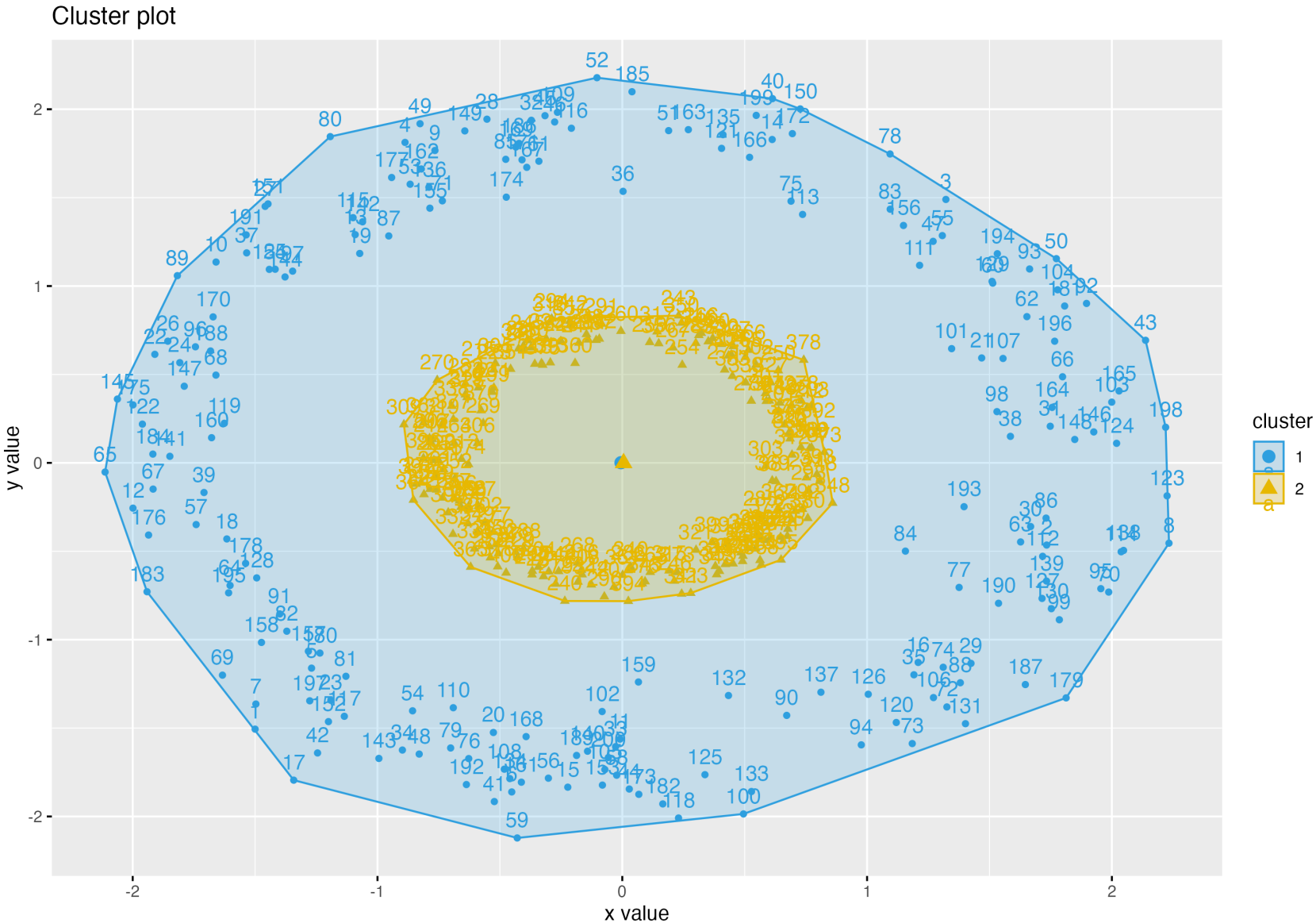
Core points — точки, от которых можем присоединять в кластер новые точки и рядом с которыми расположено `minPts` количество точек.

Пограничные точки — те, на которых кластер заканчивается.

Остальные точки считаются шумом и выбросами.

DBSCAN в R

Провести такую кластеризацию можно с помощью функции dbscan из пакета dbscan.



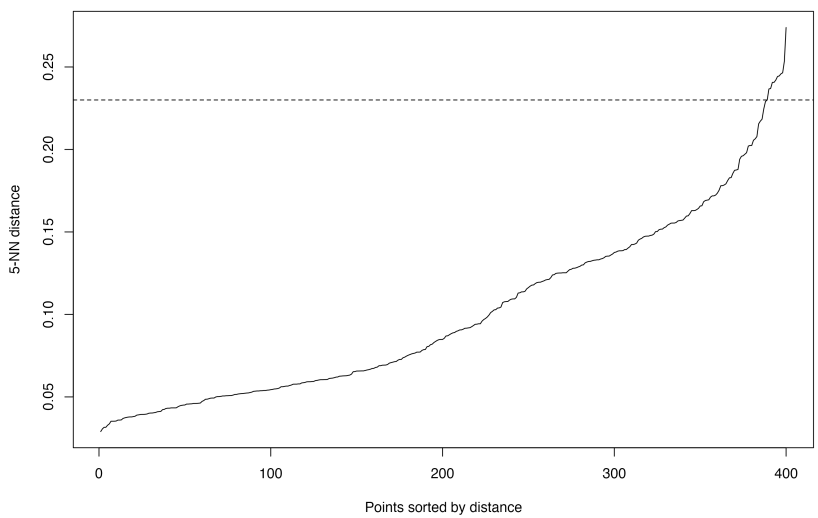
Epsilon — выбираем расстояние для радиуса

График k-расстояний (k-distance plot)

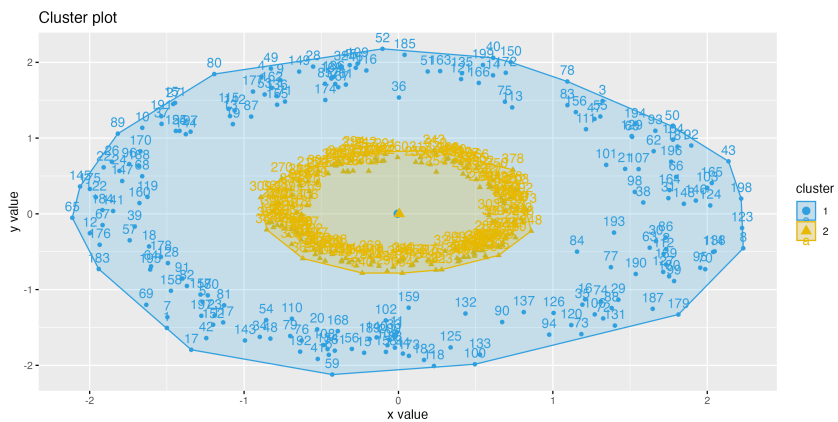
1. Вычисляется среднее значение расстояний каждой точки до ее k ближайших соседей.
2. k-расстояния отображаются в порядке возрастания.

Если на графике есть “колено” — значительный перегиб, будет легко найти нужное значение радиуса.

```
kNNdistplot(multi, k = 5)
abline(h = 0.23, lty = 2)
```




```
circle_dbscan <- dbscan(multi, 0.23,
↪  5)
fviz_cluster(circle_dbscan, data =
↪  multi,
                palette = my_col_circle)
```

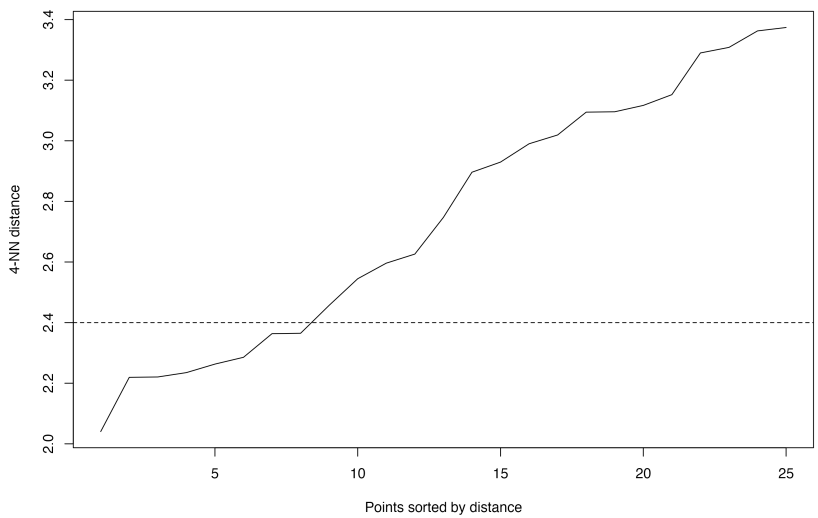


Задание 4

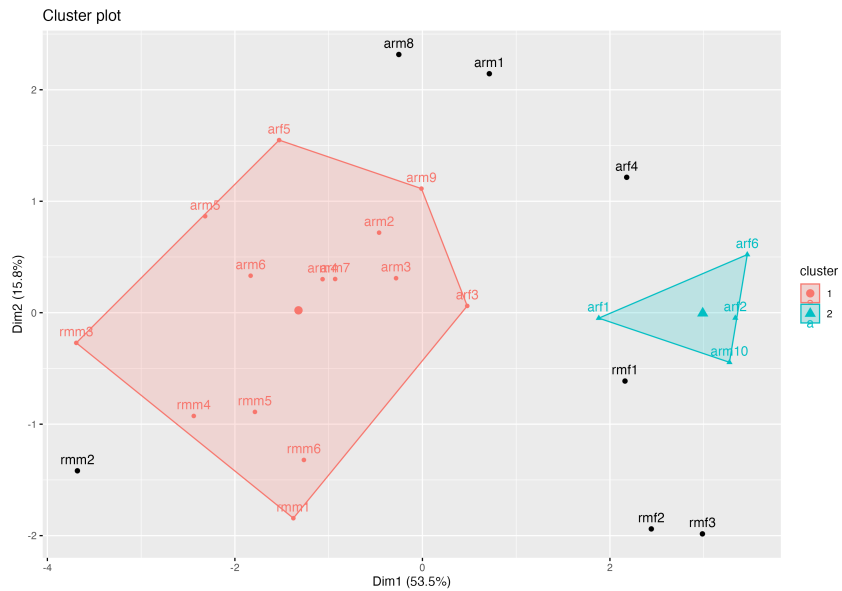
Кластеризуйте данные по волкам, используя DBSCAN-алгоритм.

Примерное решение

```
kNNdistplot(st_w, k = 4)
abline(h = 2.4, lty = 2)
```




```
w_dbscan <- dbscan(st_w, 2.4, 4)
fviz_cluster(w_dbscan, data = st_w)
```



Метод нечёткой кластеризации C-средних (C-means, fuzzy clustering)

При данном подходе объект необязательно принадлежит одному кластеру. Каждому объекту присваивается вероятность принадлежности к кластеру (membership value). В сумме все значения принадлежности к кластеру дают 1 для каждого из объектов.

Алгоритм C-means

1. Каждому объекту приписывается случайное значение принадлежности к кластеру.
2. Рассчитываются центроиды для каждого кластера:

$$\hat{v}_i = \frac{\sum_{k=1}^{N} (\hat{u}_{ik})^m y_k}{\sum_{k=1}^{N} (\hat{u}_{ik})^m}$$

где $\hat{u}$ — значение принадлежности к кластеру, m — параметр размытости (fuzziness), равный обычно 2, y_k — конкретный объект, N — количество объектов.

Алгоритм C-means

3. Расчёт расстояния от каждой точки до центроида.
4. Обновление значений принадлежности к кластерам.

$$\hat{u}_{ik} = (\sum_{j=1}^{c} (\frac{\hat{d}_{ik}}{\hat{d}_{jk}})^{\frac{2}{m-1}})^{-1}$$

где d_{ki} — расстояние от точки до центроида.

5. Повторить шаги с 2 по 4, пока не будут получены постоянные значения принадлежности к кластеру.

Кластеризация C-means в R

Есть несколько функций из разных пакетов, реализующих кластеризацию C-средних (например `fanny` из `cluster`, `cmeans` из `e1071`, `fcm` из `ppclust` и т.д.). Мы воспользуемся функцией `fanny` из пакета `cluster`.

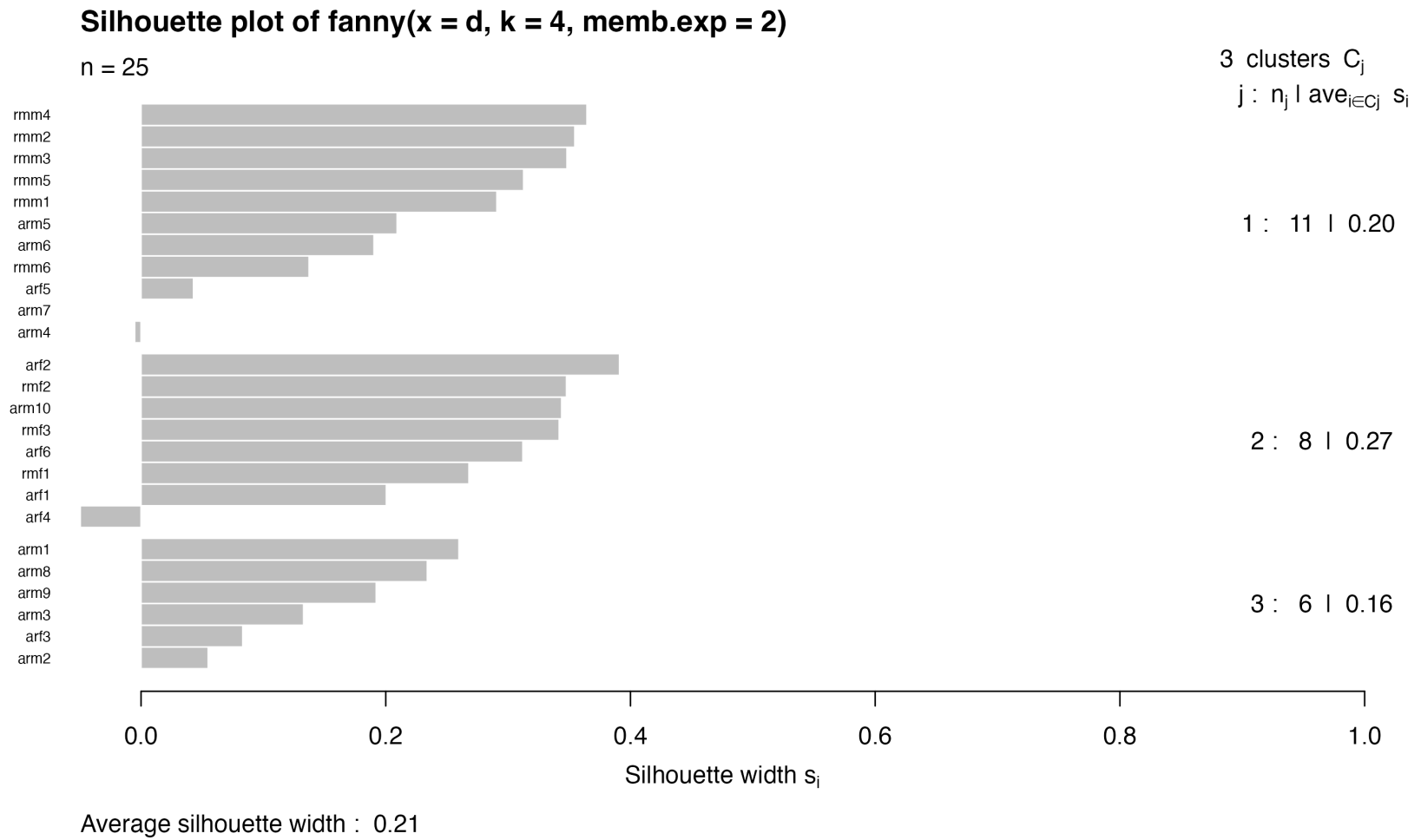
Функция funny принимает как исходные данные, так и матрицу расстояний.

```
w_cmeans <- fanny(d, k = 4, memb.exp = 2)
summary(w_cmeans)
```

```
Fuzzy Clustering object of class 'fanny' :
m.ship.expon.        2
objective      12.0661
tolerance        1e-15
iterations          85
converged            1
maxit              500
n                   25
Membership coefficients (in %, rounded):
      [,1] [,2] [,3] [,4]
rmm1    28   17   28   28
rmm2    28   16   28   28
rmm3    28   16   28   28
rmm4    29   14   29   29
rmm5    29   14   29   29
rmm6    28   17   28   28
rmf1    20   39   20   20
rmf2    21   37   21   21
rmf3    22   35   22   22
```

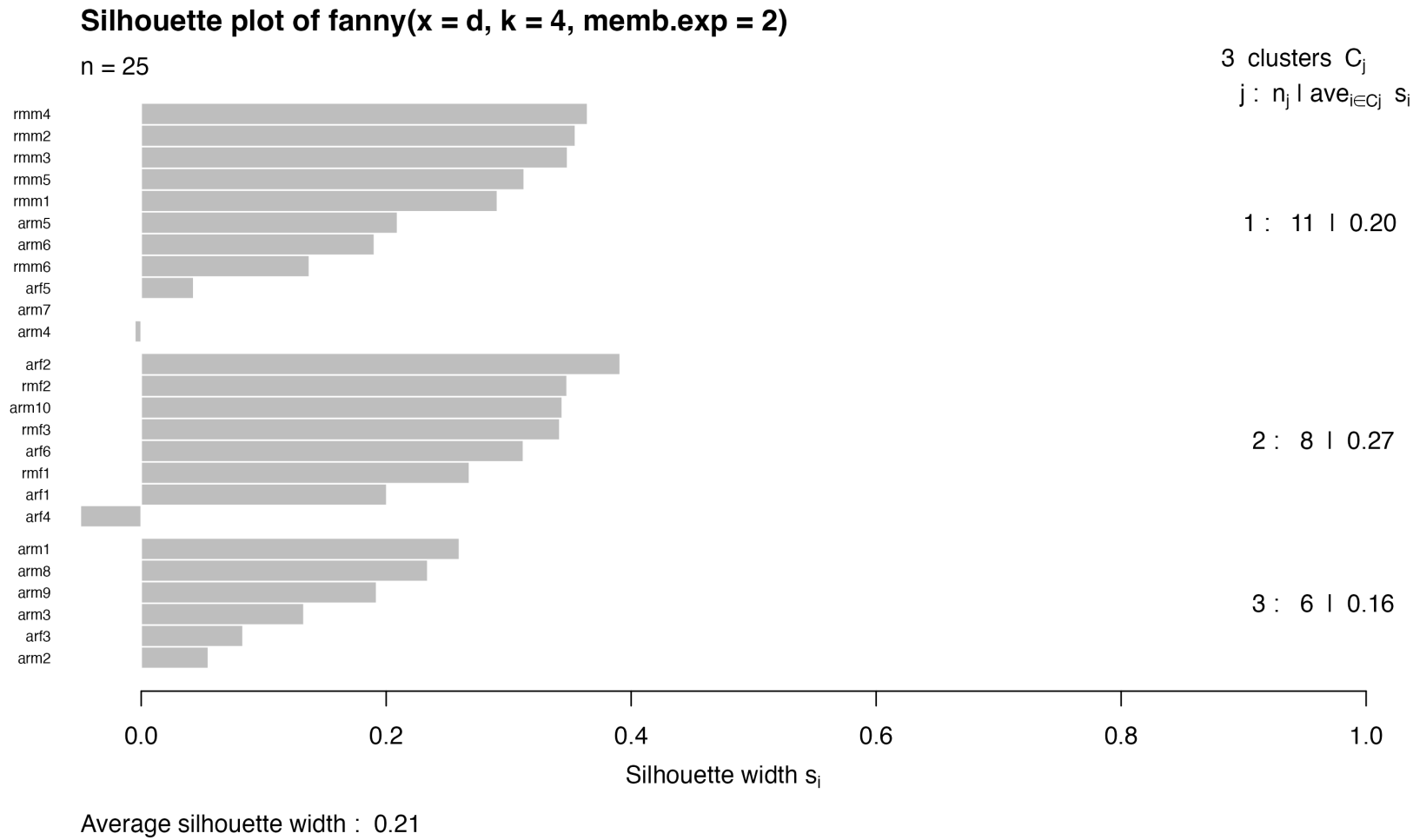
Оценка качества кластеризации: ширина силуэта

```
plot(silhouette(w_cmeans), cex.names = 0.6)
```



Оценка качества кластеризации: ширина силуэта

```
plot(silhouette(w_cmeans), cex.names = 0.6)
```



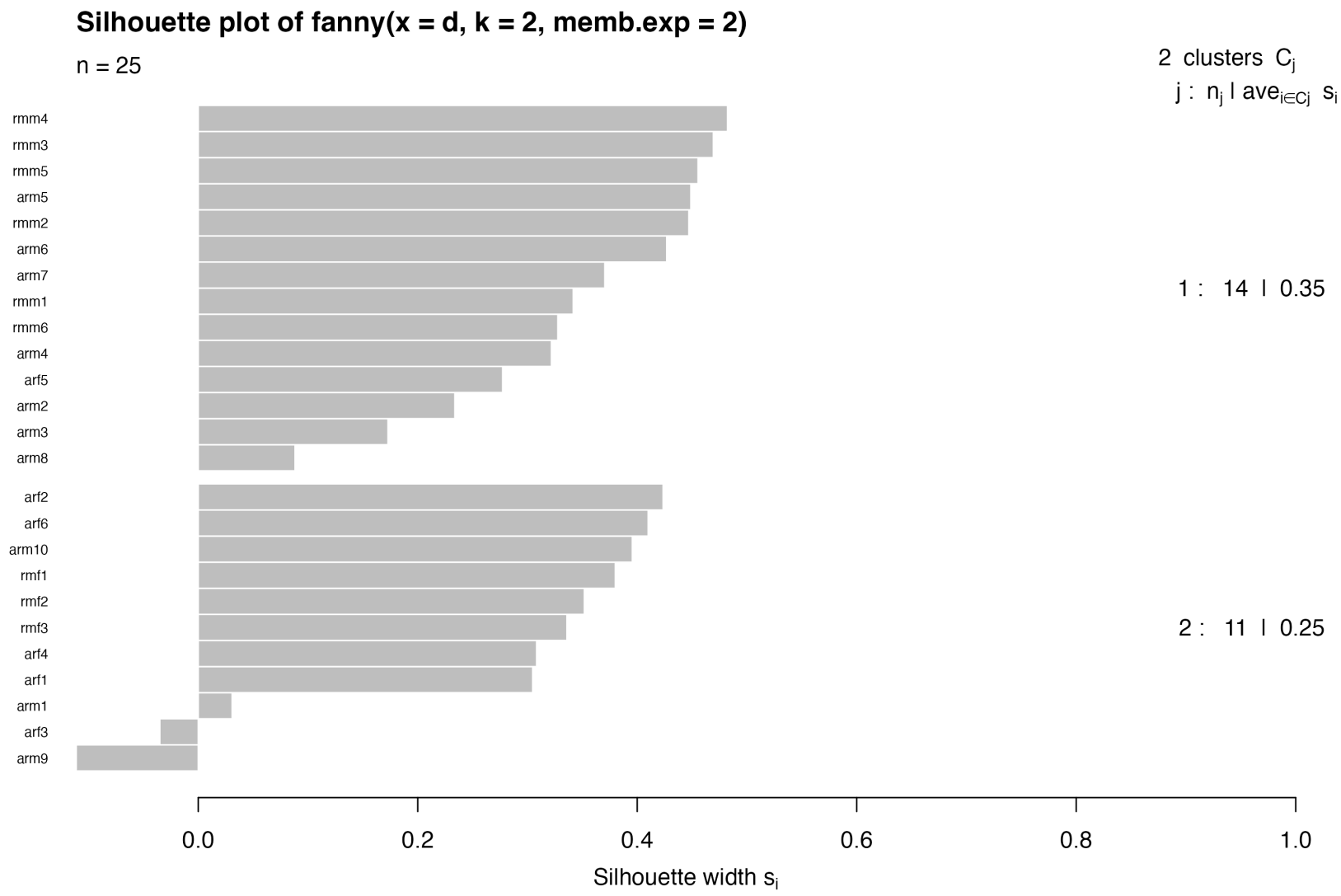
Получилась не очень качественная кластеризация.

Задание 5

Попробуйте оценить ширину силуэта для C-means кластеризации с более подходящим числом кластеров.

Решение

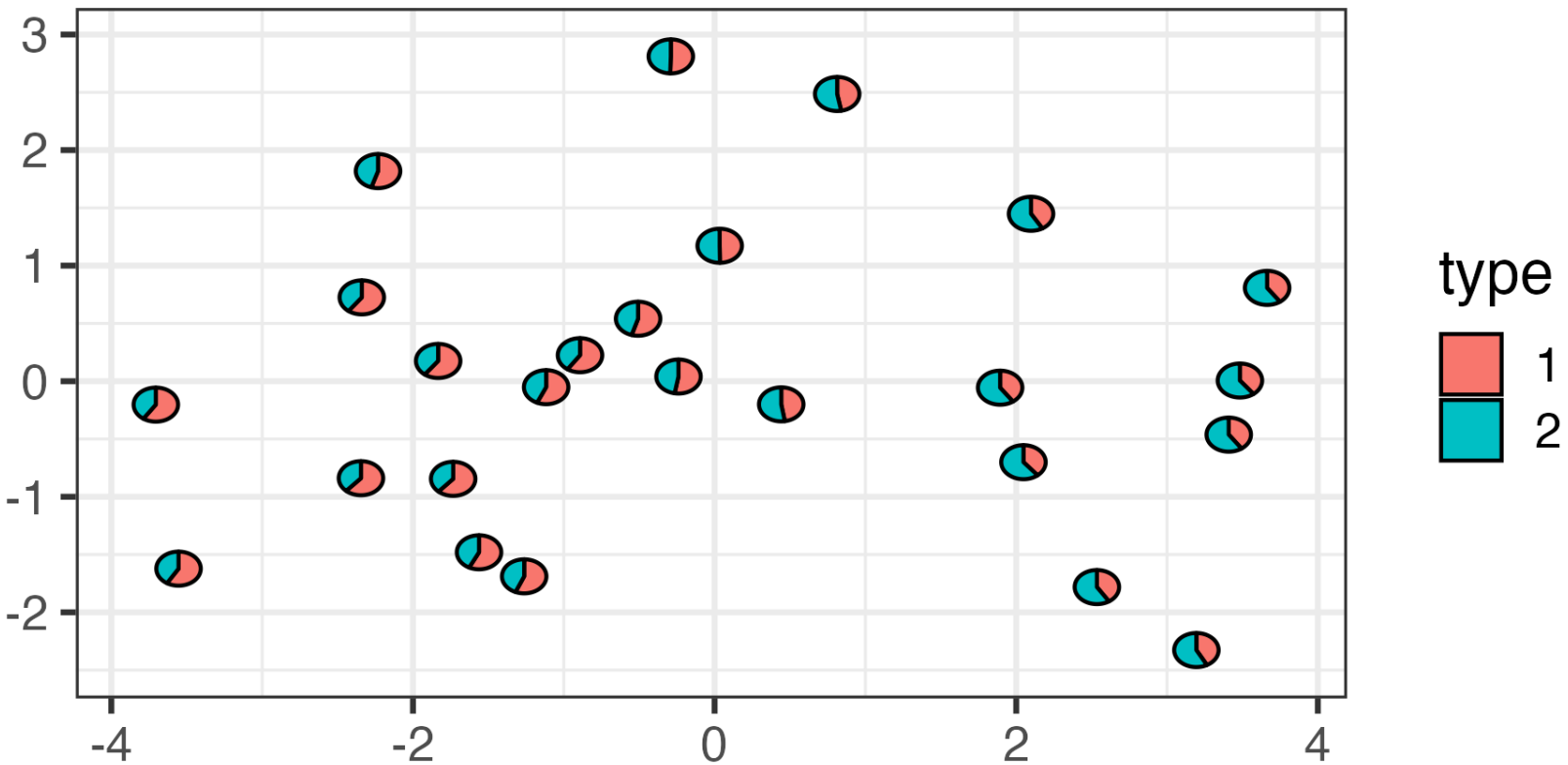
```
w_2means <- fanny(d, k = 2, memb.exp = 2)
plot(silhouette(w_2means), cex.names = 0.6)
```



Визуализация кластеризации

Для визуализации возьмём нашу исходную ординацию nMDS.

```
library(scatterpie)
w_clust <- cbind(dfr_w, w_2means$membership)
ggplot() + geom_scatterpie(data = w_clust, aes(x = MDS1, y = MDS2),
                           cols = c("1", "2"))
```



Take-home messages

- Результат кластеризации зависит не только от выбора коэффициента, но и от выбора алгоритма.
- Качество кластеризации можно оценить разными способами.
- Кластеризации, полученные разными методами, можно сравнить на танглграммах.

Дополнительные ресурсы

- Borcard, D., Gillet, F., Legendre, P., 2011. Numerical ecology with R. Springer.
- Legendre, P., Legendre, L., 2012. Numerical ecology. Elsevier.
- Quinn, G.G.P., Keough, M.J., 2002. Experimental design and data analysis for biologists. Cambridge University Press.

И еще ресурсы

Как работает UPGMA можно посмотреть здесь:

- http://www.southampton.ac.uk/~re1u06/teaching/upgma/

Как считать поддержку ветвей (пакет + статья):

- pvclust: An R package for hierarchical clustering with p-values [WWW Document], n.d. URL http://www.sigmath.es.osaka-u.ac.jp/shimo-lab/prog/pvclust/ (accessed 11.7.14).

Для анализа молекулярных данных:

- Paradis, E., 2011. Analysis of Phylogenetics and Evolution with R. Springer.

Статья про C-means кластеризацию:

- Bezdek et al., 1984. FCM: The Fuzzy c-Means Clustering Algorithm. *Computer & Geosciences, 10: 2-3*, 191-203.

Данные для самостоятельной работы

Фораминиферы маршей Белого моря (Golikova et al. 2020)

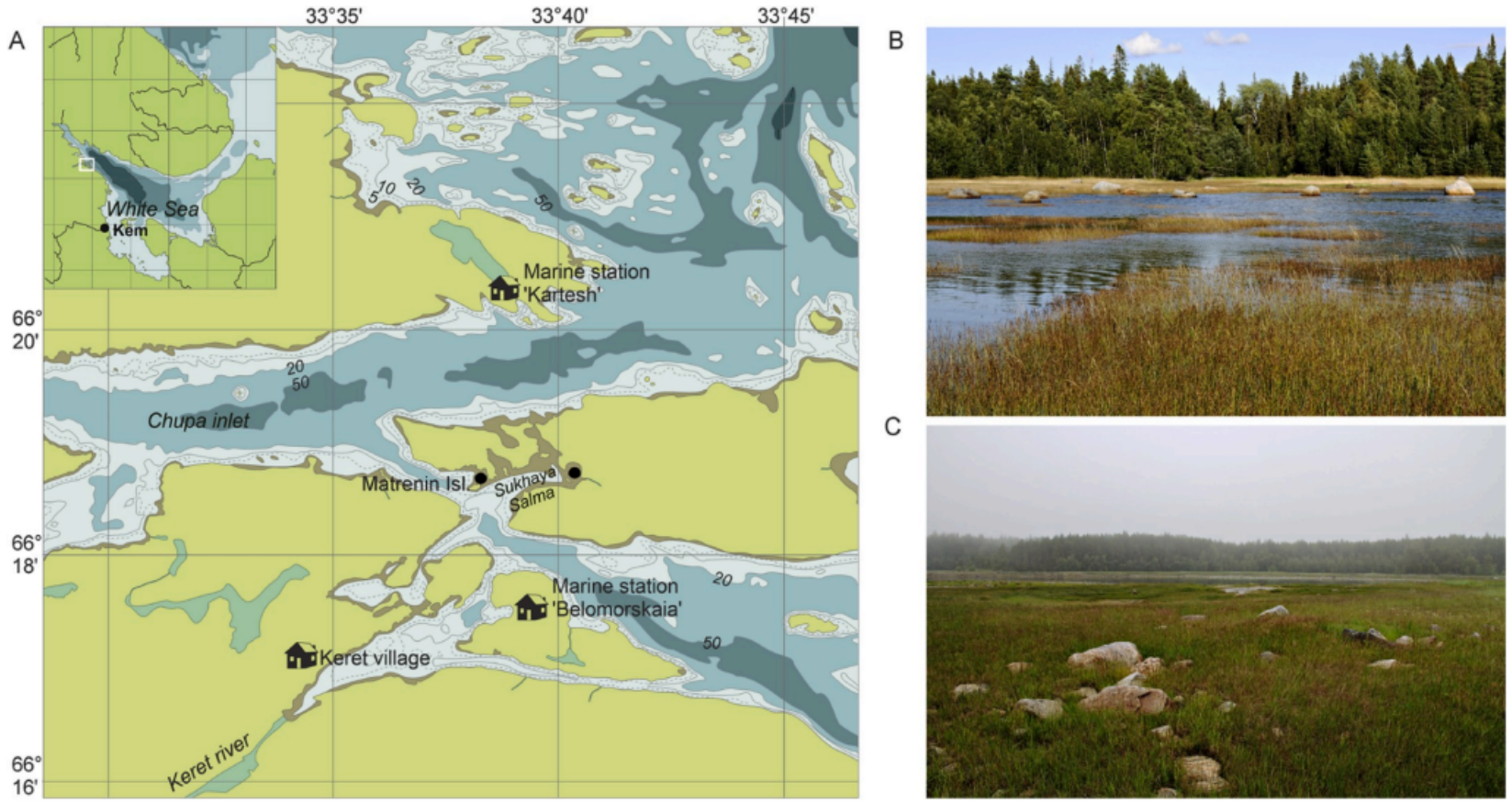


Fig. 1. A. Location of the studied saltmarshes (solid black circles) in the outer Chupa Inlet. The brown fringe marks the intertidal zone as it is shown on the nautical chart. The inset shows the White Sea with the study area boxed. **B.** Sukhaya Salma saltmarsh, high tide. **C.** Matrenin saltmarsh, low tide. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Фораминиферы маршей Белого моря

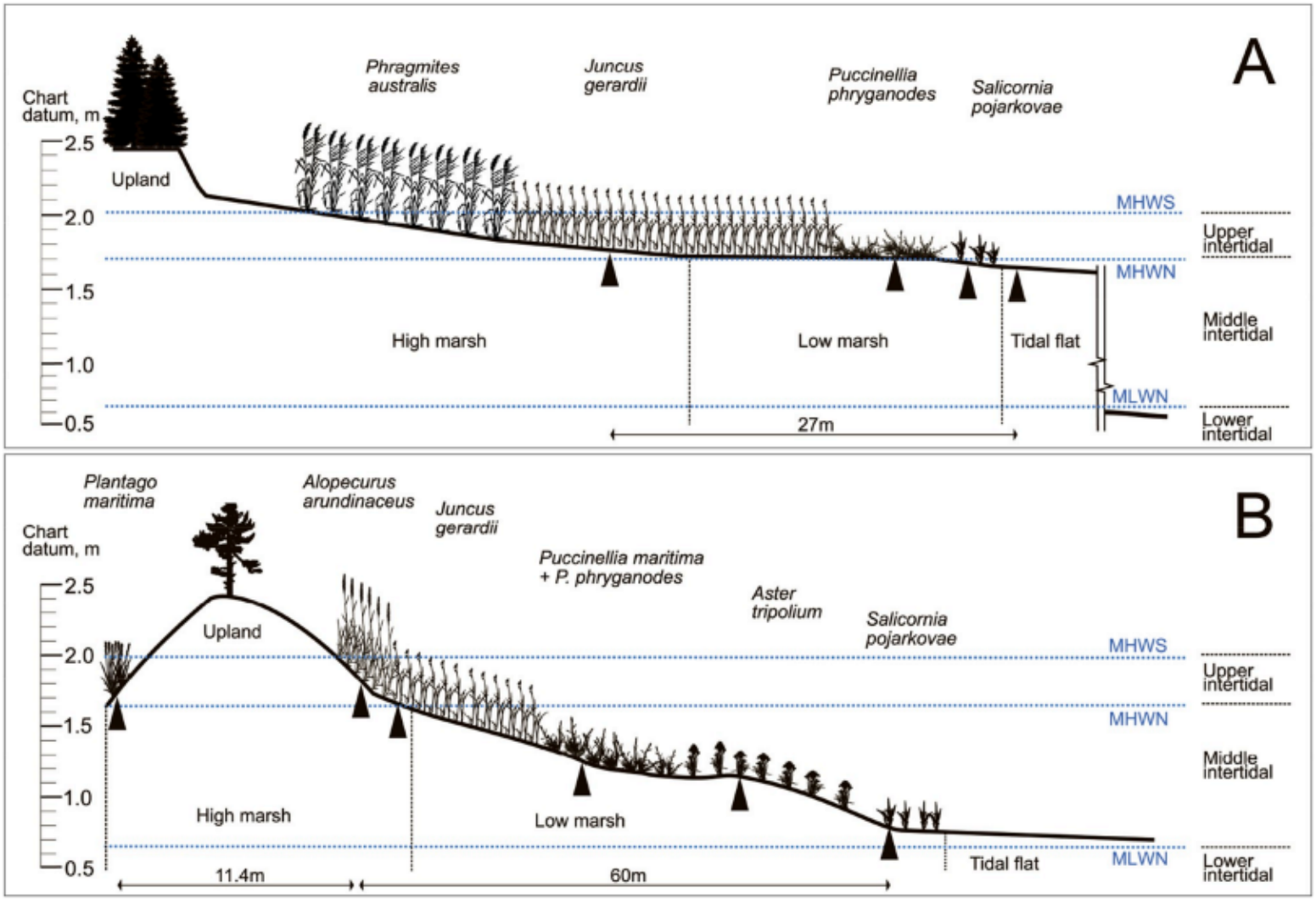


Fig. 2. Schematic transects of the White Sea saltmarshes. A. Sukhaya Salma. **B.** Matrenin. Vegetation belts are labeled above the transects. Arrowheads are foraminiferal sampling stations. Regional tidal levels calculated with WXTide32 are shown in blue color: MHWS mean high water at spring tides, MHWN mean high water at neap tides, MLWN mean low water at neap tides. The boundary between the high marsh and low marsh is MHWN. Intertidal zones are labeled on the right. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Фораминиферы маршей Белого моря

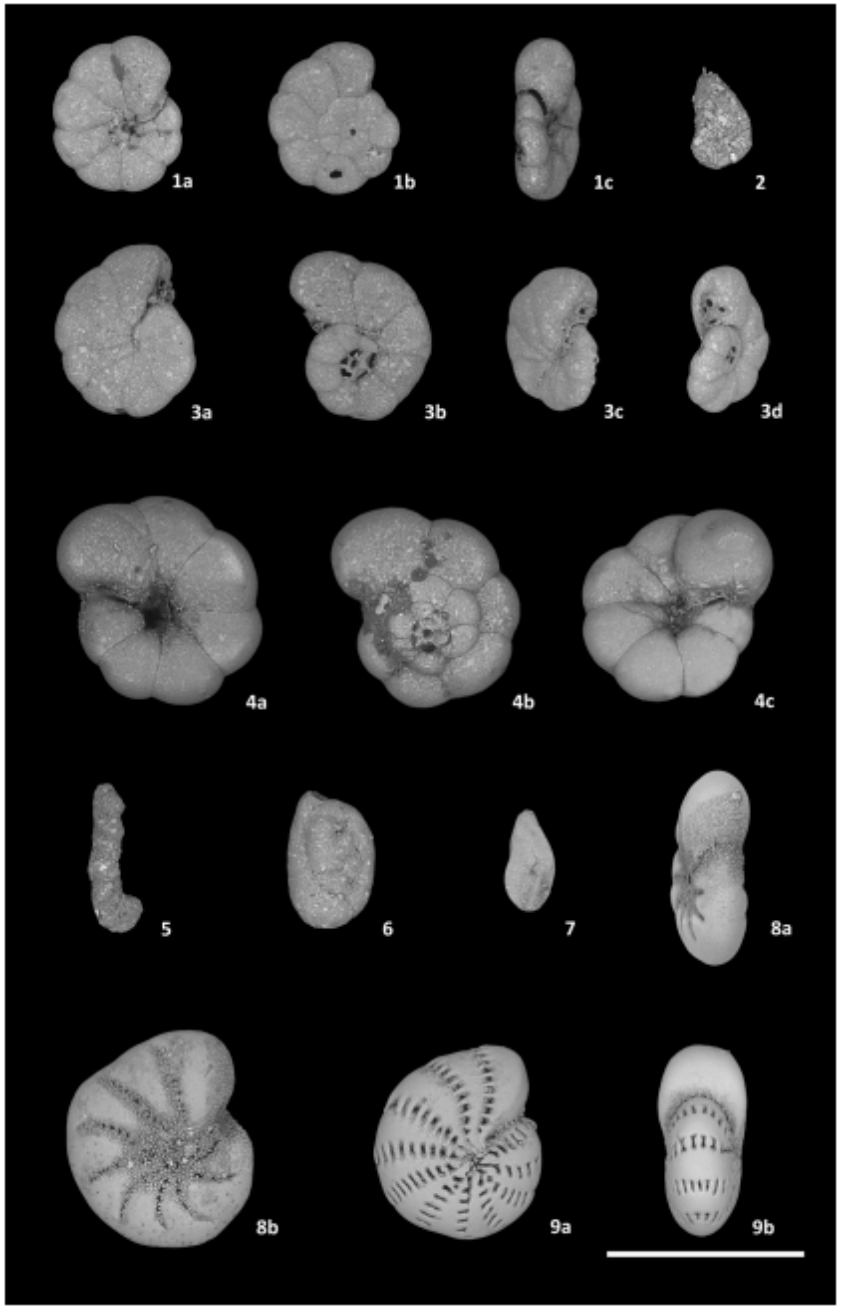


Plate 1.

1. *Balticammina pseudomacrescens*.
2. *Ammotium salsum*.
3. *Jadammina macrescens*.
4. *Trochammina inflata*.
5. *Ammobaculites balkwilli?*
6. *Miliammina fusca*.
7. *Ovammina opaca*.
8. *Elphidium albiumbilicatum*.
9. *Elphidium williamsoni*.

Scale bar 500 µm.

Задание

Проанализируйте данные об относительных обилиях фораминифер в пробах на Белом море.

- Выберите и обоснуйте трансформацию данных и расстояние.
- Постройте ординацию nMDS по относительным обилиям фораминифер:
  - цвет значков — растение-доминант,
  - форма значков — точка сбора.
- Постройте дендрограмму проб по сходству относительных обилий фораминифер.
  - оцените при помощи кофенетической корреляции, какой метод аггрегации лучше,
- Постройте визуализацию для методов неиерархической кластеризации
- Опишите получившиеся кластеры при помощи различных параметров:
  - ширина силуэта
  - бутстреп-поддержка ветвлений

Фораминиферы маршей Белого моря

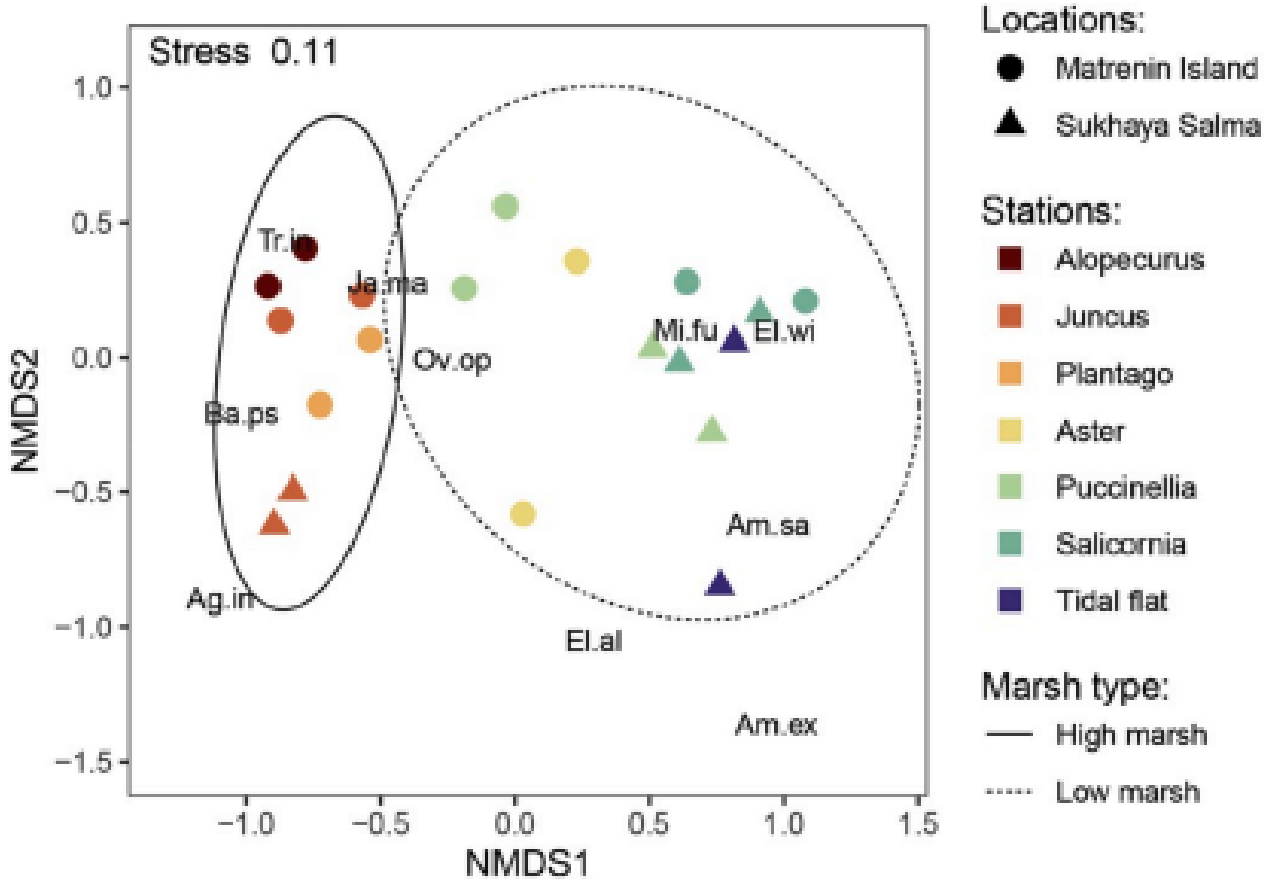


Fig. 5. Ordination of foraminiferal assemblages using the nonmetric multidimensional scaling. Distances between points are proportional to Brey-Curtis dissimilarities. Stress value estimates the goodness of fit. Shapes code location; color codes vegetation belts. The abbreviations in the plot stand for foraminiferal species; their position indicates the association of abundances with vegetation. 95% confidence ellipses are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Фораминиферы маршей Белого моря

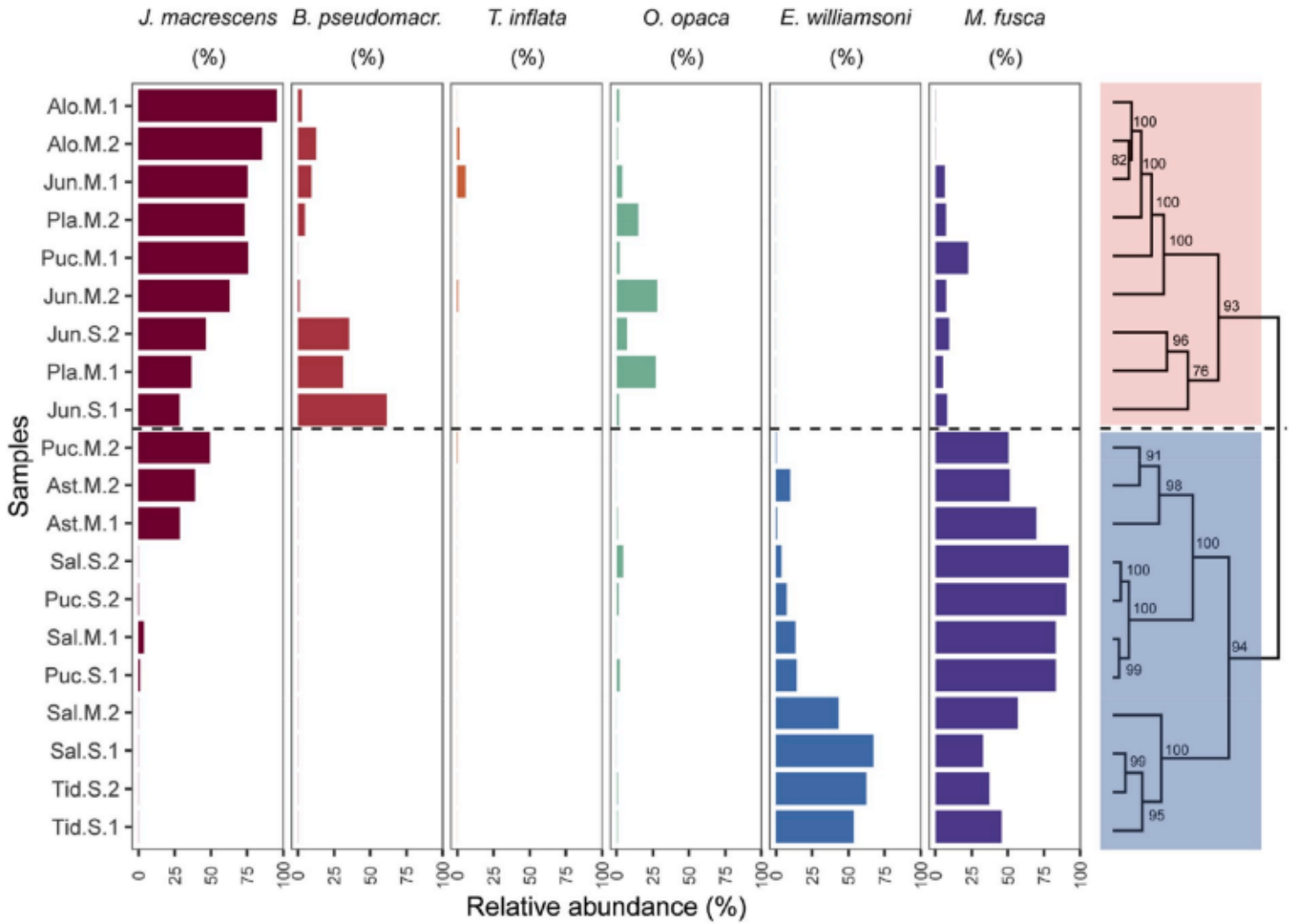


Fig. 6. Clusterization of samples, based on relative occurrence of living foraminifera. Strength of the cluster support by data is expressed in approximately unbiased p-values (AU p-values). The stations are arranged according to the cluster analysis results (station labels: plant species_saltmarsh_replicate). Pink codes the high marsh stations (upper cluster), blue – low marsh (lower cluster). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)