Министерство образования и науки РФ
ФГБОУ ВО «Кубанский государственный технологический университет»
Институт компьютерных систем и информационной безопасности
Кафедра информатики и вычислительной техники

Отчет

По лабораторной работе № 3
По дисциплине анализ и визуализация данных

Выполнил студент группы 19-КМ-ИВ1: Кирмасов Б.В.

Преподаватель: Решетняк М.Г.

Краснодар 2020

Тема: «Кластеризация данных».
Цель работы: «Научиться выполнять кластеризацию данных в R с помощью методов KMeans, иерархического, смеси гауссовских распределений».

Отчёт о работе.

```
# Image Posterization------------------------------------------------------
---

library(jpeg)
library(reshape)
library(ggplot2)
install.packages("reshape")

# Part 1
# Загрузка изображения


imageLoaderLocal <- function(nm){
  readImage <- readJPEG(sprintf("img_R/%s.jpg", nm))

  longImage <- melt(readImage)
  rgbImage <- reshape(longImage, timevar = "X3",
                      idvar = c("X1", "X2"), direction = "wide")
  rgbImage$X1 <- -rgbImage$X1
  return(rgbImage)
}

# Part 2
# Идентификация "доминирующих" цветов с помощью k-средних
# Помещаем изображение в rgbImage
# rgbImage <- imageLoader(allImageURLs[2]) # Pick one, or use your own URL.
rgbImage <- imageLoaderLocal(1)
with(rgbImage, plot(X2, X1, col = rgb(rgbImage[, 3:5]), asp = 1, pch = "."))
```

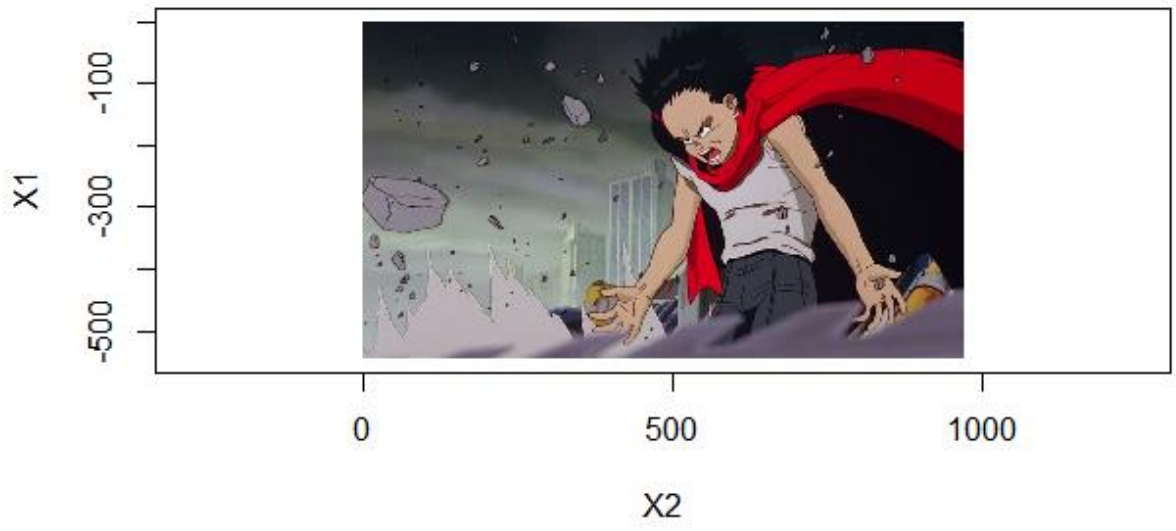

Рисунок 1 – Изображение в rgbImage

```
# Выполняем кластеризацию пикселей изображения на указанное в kColors количес
тво групп
kColors <- 3 # Number of palette colors
kMeans <- kmeans(rgbImage[, 3:5], centers = kColors)

# Просмотр данных кластеров
kMeans
factor(kMeans$cluster)

# k доминирующие цвета
zp1 <- qplot(factor(kMeans$cluster), geom = "bar",
             fill = factor(kMeans$cluster))
zp1 <- zp1 + scale_fill_manual(values = rgb(kMeans$centers))
zp1
```

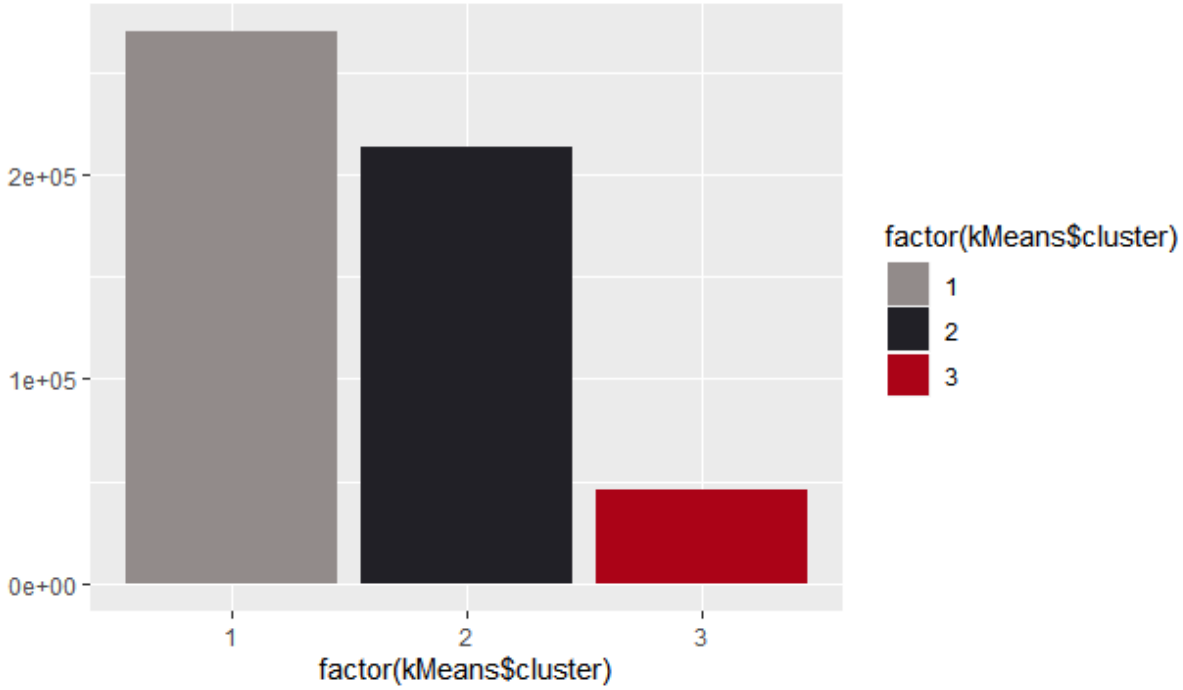

Рисунок 2 – Доминирующие цвета

```
# Постеризация
approximateColor <- kMeans$centers[kMeans$cluster, ]
with(rgbImage, plot(X2, X1, col = rgb(approximateColor), asp = 1, pch = "."))
```

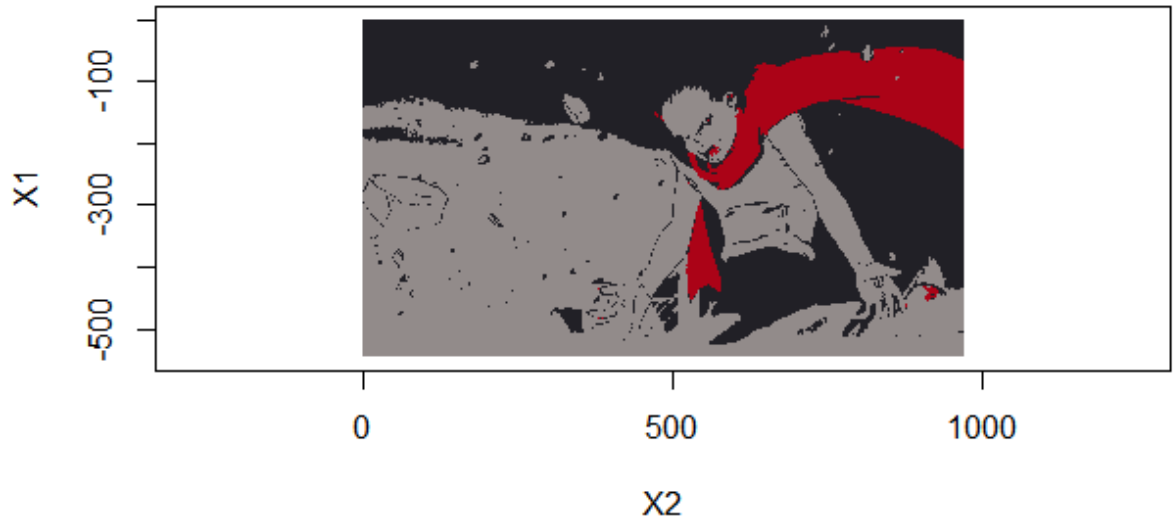

Рисунок 3 – Постеризованное изображение

```
# Task
# Как сделать, чтобы постеризация не изменялась?
# Нужно задать значения цветов в rgb(), либо сделать кластеризацию с большим
количеством цветов палитры (сейчас стоит kColors <- 3, нужно поставить, напри
мер, 100)


# K-Means vs Hierarchical clustering-------------------------------------------
---

library(cluster)

# Part 1
# Wine

# data(wine, package='rattle')
# wine
# saveRDS(wine, "Data/wine.RData")

# Загружаем в wine данные из wine.RData
wine <- readRDS("wine.RData")

# Стандартизация переменных
wine.stand <- scale(wine[-1])

# K-Means
k.means.fit <- kmeans(wine.stand, 3) # k = 3
attributes(k.means.fit)

# Centroids:
k.means.fit$centers

# Clusters:
k.means.fit$cluster

# Cluster size:
k.means.fit$size

# Находим сумму квадратов внутри группы
wssplot <- function(data, nc = 15, seed = 1234) {
  wss <- (nrow(data) - 1) * sum(apply(data, 2, var))
  for (i in 2:nc) {
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)
```

```
  }
  plot(1:nc, wss, type = "b", xlab = "Number of Clusters",
       ylab = "Within groups sum of squares")
}

# Выводим график
wssplot(wine.stand, nc = 10)

# Task
# Исходя из графика, какое оптимальное количество кластеров и почему?
# Если построить график зависимости внутрикластерной суммы квадратов от колич
ества кластеров, первые кластеры будут добавлять много информации (объяснять
большую долю дисперсии). Но в некоторой точке предельная выгода (усиление мод
ели) начнет снижаться, что отразится в появлении точки перегиба на графике, –
так называемый “elbow criterion”. Число кластеров выбирается в этой точке.
# 2-4 кластера, так как если брать кластеров > 4, то разница суммы квадратов
стремится к минимуму

# 2D-представление кластерного решения
clusplot(wine.stand, k.means.fit$cluster,
         main='2D representation of the Cluster solution',
         color=TRUE, shade=TRUE, labels=2, lines=0)
```

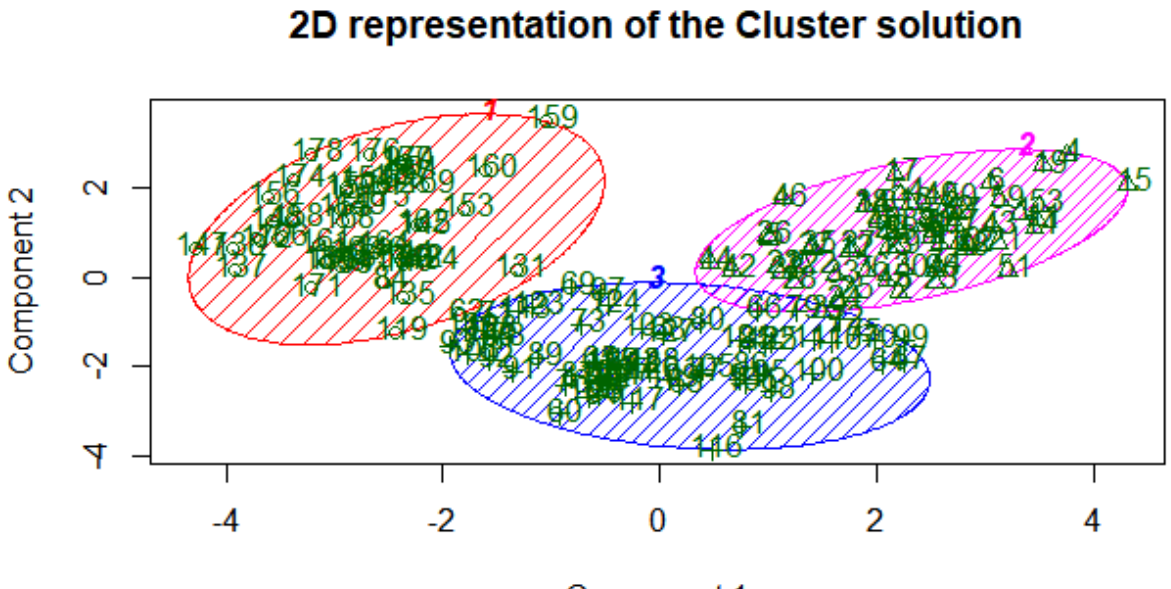

Рисунок 4 – 2D-представление кластерного решения

```
table(wine[, 1], k.means.fit$cluster)

     1  2  3
  1  0 59  0
  2  3  3 65
  3 48  0  0

# Task
# Что изображено на графике?
# На графике изображены значения, разбитые по кластерам
# О чём говорят данные таблицы?
# Для оценки эффективности кластеризации строим матрицу несоответствий (confu
sion matrix). Матрица показывает, что все объекты 1 класса были отнесены ко 2
классу, из всех объектов 2 класса правильно было отнесено только 3 объекта, и
все объекты 3 класса были ошибочно отнесены к 1 классу.

# Hierarchical clustering
# Методы иерархической кластеризации используют матрицу расстояний в качестве
входных данных для алгоритма кластеризации. Выбор подходящей метрики будет
влиять на форму кластеров, так как некоторые элементы могут быть близки друг
к другу, в соответствии с одним расстоянием и дальше в соответствии с другим.
d <- dist(wine.stand, method = "euclidean") # Euclidean distance matrix.
```

```
H.fit <- hclust(d, method = "ward.D2")

plot(H.fit) # display dendogram
# Результат кластеризации может быть представлен в виде дендрограммы.
```

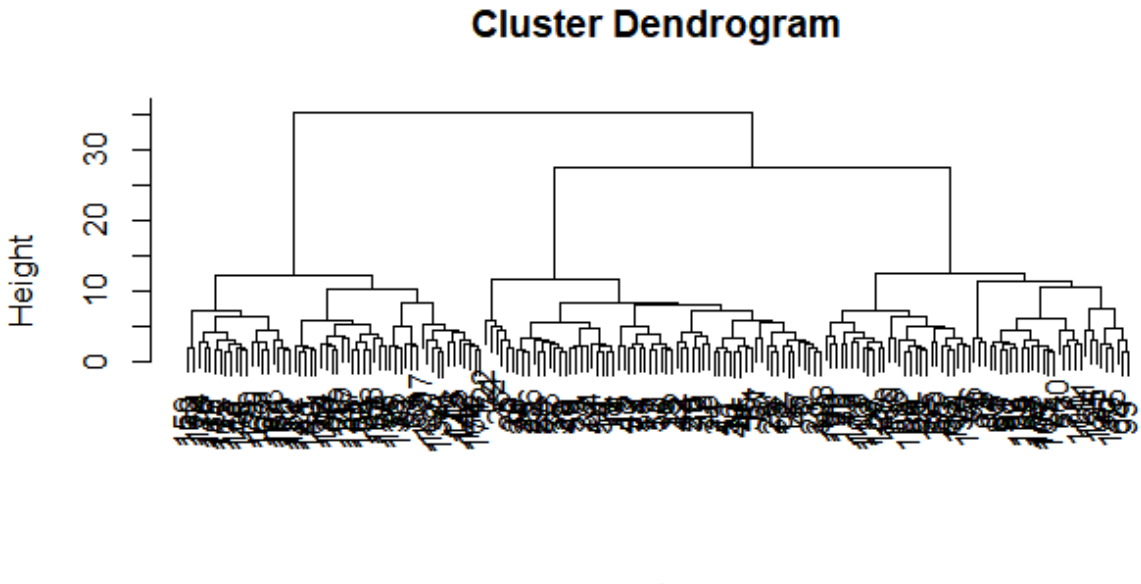

Рисунок 5 - Дендрограмма

```
groups <- cutree(H.fit, k = 3) # делим дерево на 3 группы

# Нарисуем дендрограмму с красными границами вокруг групп
rect.hclust(H.fit, k = 3, border = "red")
```

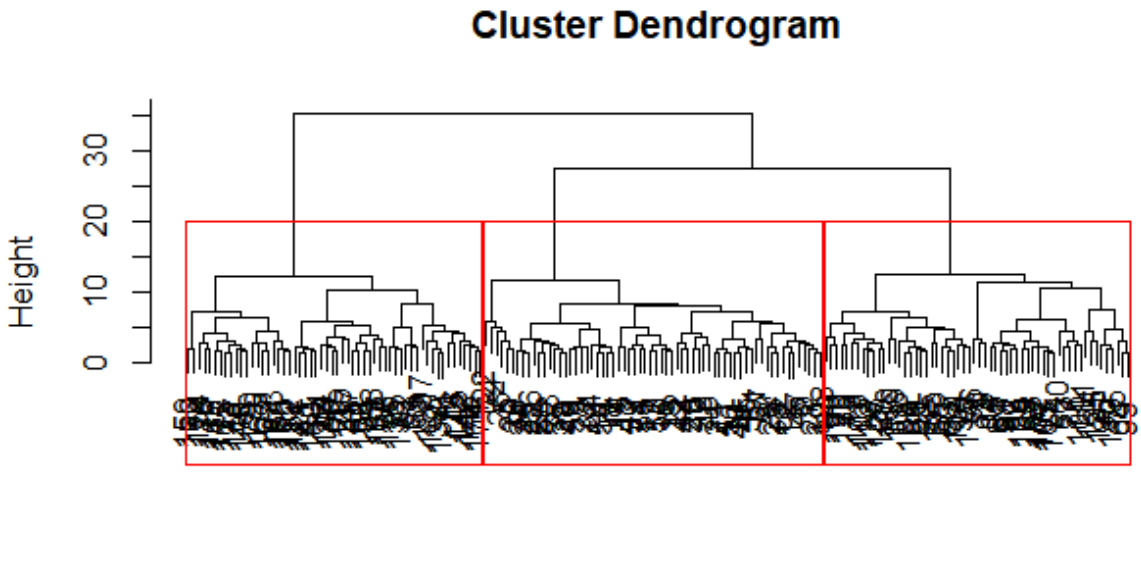

Рисунок 5 – Дендрограмма с красными границами

```
table(wine[, 1], groups)

groups
     1  2  3
  1 59  0  0
  2  5 58  8
  3  0  0 48


# Part 2
# Protein
# Мы рассматриваем 25 европейских стран (n = 25 единиц) и их потребление белк
а (в процентах) из девяти основных источников пищи (p = 9).
food <- read.csv("protein.csv")

set.seed(123456789) # чтобы исправить случайные стартовые кластеры

# K-Means
# Сначала мы начнем с кластеризации только красного и белого мяса (p = 2) и k
= 3 кластера.
grpMeat <- kmeans(food[ , c("WhiteMeat", "RedMeat")], centers = 3, nstart = 1
0)
grpMeat

K-means clustering with 3 clusters of sizes 12, 5, 8

Cluster means:
  WhiteMeat   RedMeat
1  4.658333  8.258333
2  9.000000 15.180000
3 12.062500  8.837500

Clustering vector:
 [1] 1 3 2 1 3 3 3 1 2 1 3 2 1 3 1 3 1 1 1 1 2 2 1 3 1

Within cluster sum of squares by cluster:
[1] 69.85833 35.66800 39.45750
 (between_SS / total_SS =  75.7 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss
"
[6] "betweenss"    "size"         "iter"         "ifault"


# Сортировка
o <- order(grpMeat$cluster)
data.frame(food$Country[o], grpMeat$cluster[o])

food.Country.o. grpMeat.cluster.o.
1          Albania                  1
2         Bulgaria                  1
3          Finland                  1
4           Greece                  1
5            Italy                  1
6           Norway                  1
7         Portugal                  1
8          Romania                  1
9            Spain                  1
10          Sweden                  1
11            USSR                  1
12      Yugoslavia                  1
13         Belgium                  2
14          France                  2
15         Ireland                  2
16     Switzerland                  2
17              UK                  2
18         Austria                  3
19  Czechoslovakia                  3
20         Denmark                  3
21       E Germany                  3
```

```
22         Hungary                  3
23     Netherlands                  3
24          Poland                  3
25       W Germany                  3
```

```
# Чтобы увидеть графическое представление решения кластеризации, мы наносим к
ластерные назначения на красное и белое мясо на диаграмме рассеяния:
# Makes plot
plot(food$RedMeat, food$WhiteMeat, type = "n", xlim = c(3, 19),
     xlab = "Red Meat", ylab = "White Meat")
text(x = food$RedMeat, y = food$WhiteMeat,
     labels = food$Country, col = grpMeat$cluster + 1)
```

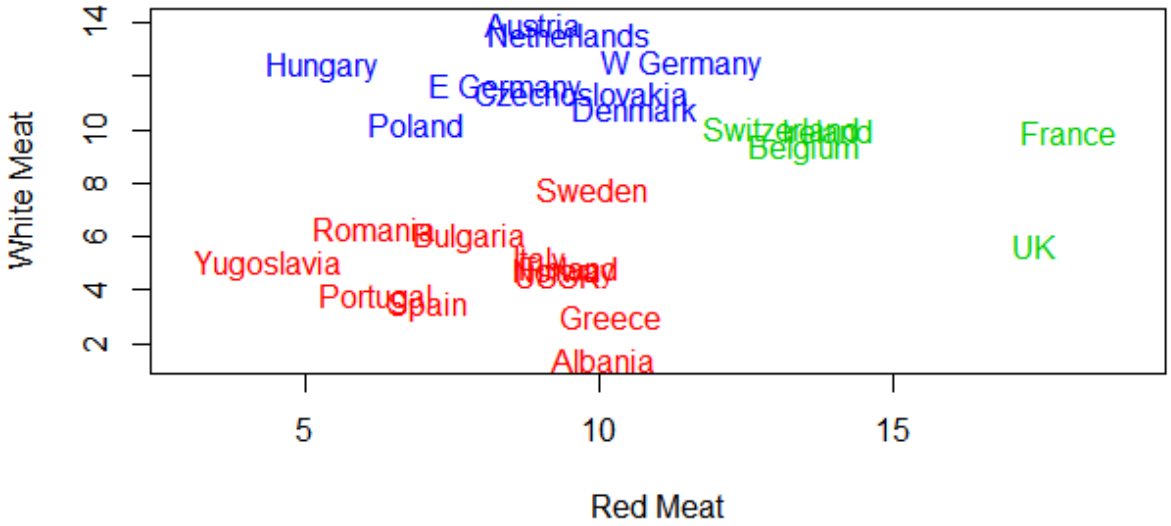

Рисунок 6 – Диаграмма рассеяния красного и белого мяса

```
# Затем мы объединяем все девять групп белков и готовим программу для создани
я семи кластеров. Получающиеся группы, показанные в цвете на диаграмме рассея
ния белого мяса против красного мяса (любая другая пара особенностей могла бы
быть выбрана), фактически имеет много смысла. Страны, находящиеся в тесной ге
ографической близости, как правило, объединяются в одну группу.
```

```
# Same analysis, but now with clustering on all protein groups
# change the number of clusters to 7
grpProtein <- kmeans(food[ , -1], centers = 7, nstart = 10)
o <- order(grpProtein$cluster)
data.frame(food$Country[o], grpProtein$cluster[o])
```

```
   food.Country.o. grpProtein.cluster.o.
1         Bulgaria                      1
2          Romania                      1
3       Yugoslavia                      1
4         Portugal                      2
5            Spain                      2
6          Denmark                      3
7          Finland                      3
8           Norway                      3
9           Sweden                      3
10         Belgium                      4
11          France                      4
12         Ireland                      4
13     Switzerland                      4
14              UK                      4
15         Austria                      5
16       E Germany                      5
17     Netherlands                      5
```

```
18       W Germany                    5
19  Czechoslovakia                    6
20         Hungary                    6
21          Poland                    6
22         Albania                    7
23          Greece                    7
24           Italy                    7
25            USSR                    7
```

```
clusplot(food[ , -1], grpProtein$cluster,
         main = '2D representation of the Cluster solution',
         color = TRUE, shade = TRUE,
         labels = 2, lines = 0)
```

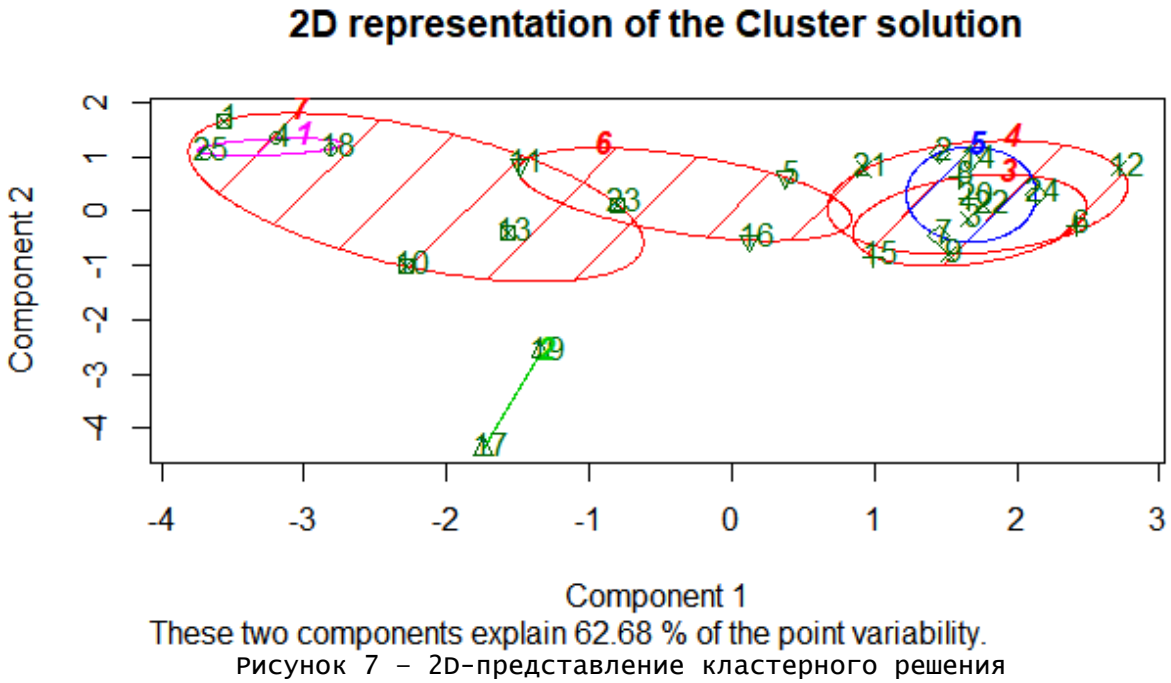

Рисунок 7 – 2D-представление кластерного решения

```
# Иерархическая кластеризация
# В качестве альтернативы мы можем реализовать иерархический подход. Мы испол
ьзуем функцию agnes в кластере пакетов. Аргумент дисс = ЛОЖЬ указывает, что м
ы используем матрицу различий, которая рассчитывается на основе необработанны
х данных. Аргумент метрика = «евклидов» указывает, что мы используем евклидов
о расстояние. Стандартизация не используется, а функция связи является «средн
ей» связью.
foodagg <- agnes(food, diss = FALSE, metric = "euclidian")
plot(foodagg, main = "Dendrogram")
```

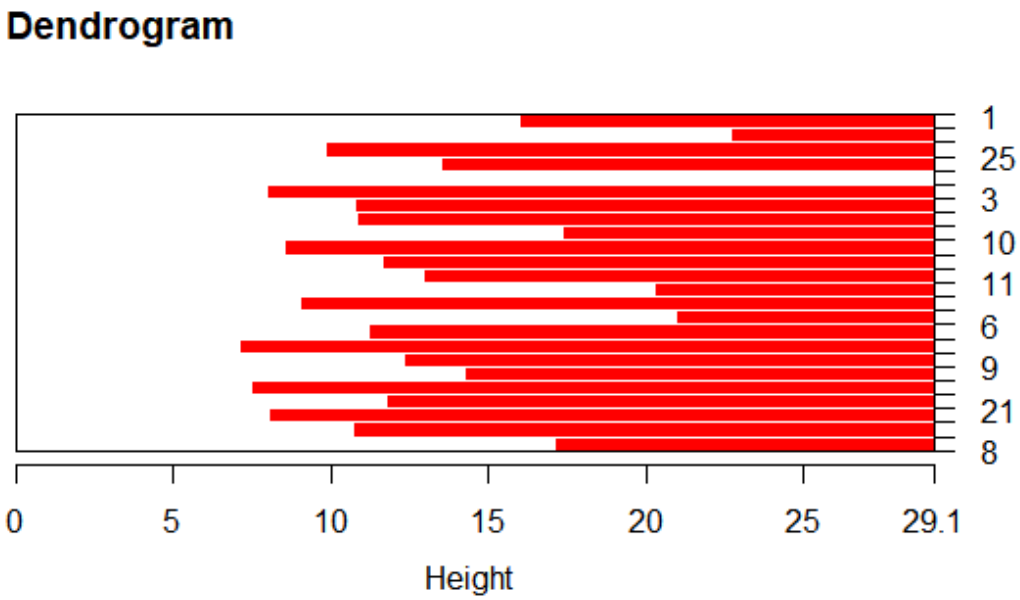

Рисунок 8 – Дендрограмма

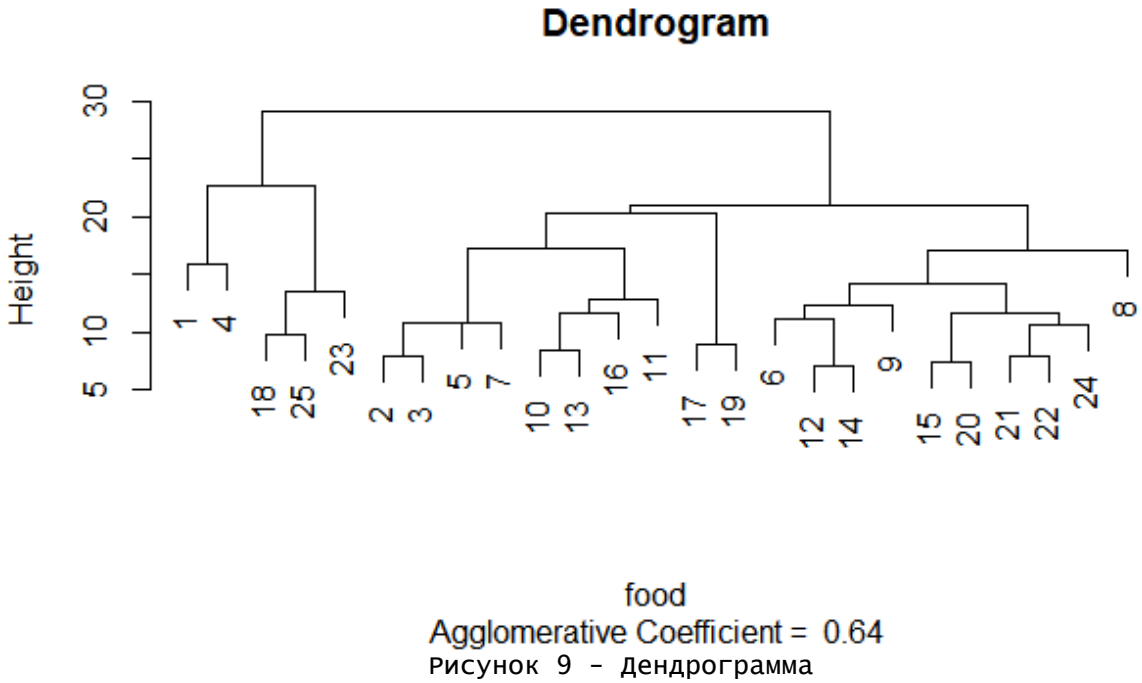

Рисунок 9 – Дендрограмма

```
groups <- cutree(foodagg, k = 4) # cut tree into 4 clusters
rect.hclust(foodagg, k = 4, border = "red")
```

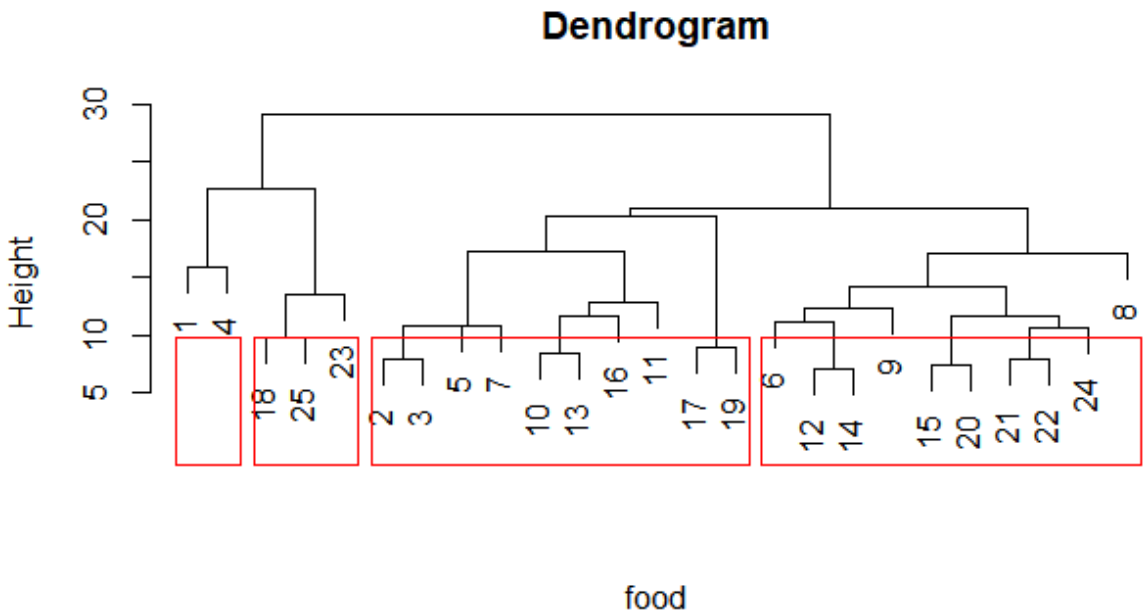

Рисунок 10 – Дендрограмма с красными границами кластеров

```
# Gaussian mixture models-------------------------------------------------
---

library(mclust)

# Загружаем в df встроенный набор данных
df <- mtcars[, c("mpg", "hp")]

# Количество кластеров определяется моделью
fit <- Mclust(df)
plot(fit, what = "classification")
```

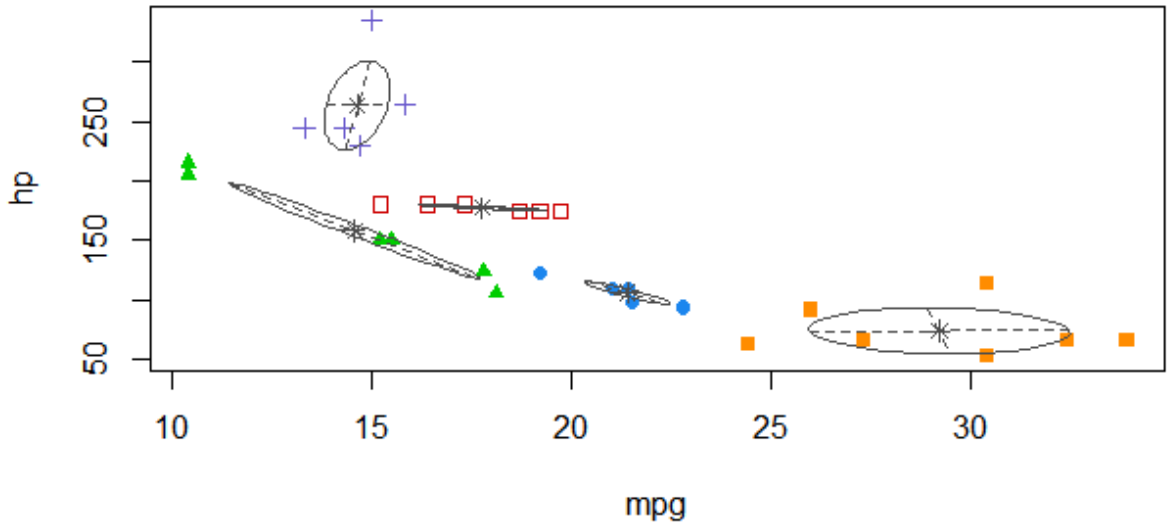

Рисунок 11

```
# Выводим атрибуты
attributes(fit)

$names
 [1] "call"           "data"           "modelName"      "n"
 [5] "d"              "G"              "BIC"            "bic"
 [9] "loglik"         "df"             "hypvol"         "parameters"
[13] "z"              "classification" "uncertainty"

$class
[1] "Mclust"

# Выводим модель
summary(fit, parameter = TRUE)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 5
components:

 log-likelihood  n df      BIC       ICL
      -217.9743 32 29 -536.455 -536.5947

Clustering table:
1 2 3 4 5
8 6 6 5 7

Mixing probabilities:
        1         2         3         4         5
0.2482399 0.1872860 0.1873787 0.1565017 0.2205937

Means:
         [,1]     [,2]      [,3]      [,4]     [,5]
mpg  21.38272  17.7529  14.56497  14.62096 29.19549
hp  105.94065 177.4971 158.02444 263.65808 73.76783

Variances:
[,,1]
          mpg       hp
mpg  1.137669 -9.39726
hp  -9.397260 89.50181
[,,2]
          mpg        hp
mpg  2.551351 -3.621873
hp  -3.621873  6.249992
[,,3]
            mpg        hp
mpg    9.823005 -124.3821
hp  -124.382092 1603.1666
[,,4]
           mpg         hp
mpg  0.6771384   10.98042
hp  10.9804195 1394.22300
[,,5]
          mpg         hp
mpg 10.671079  -5.565029
hp  -5.565029 379.570735


# Одна переменная
# Число кластеров 5
fit <- Mclust(df$mpg, G = 5)
plot(fit, what = "classification")
```

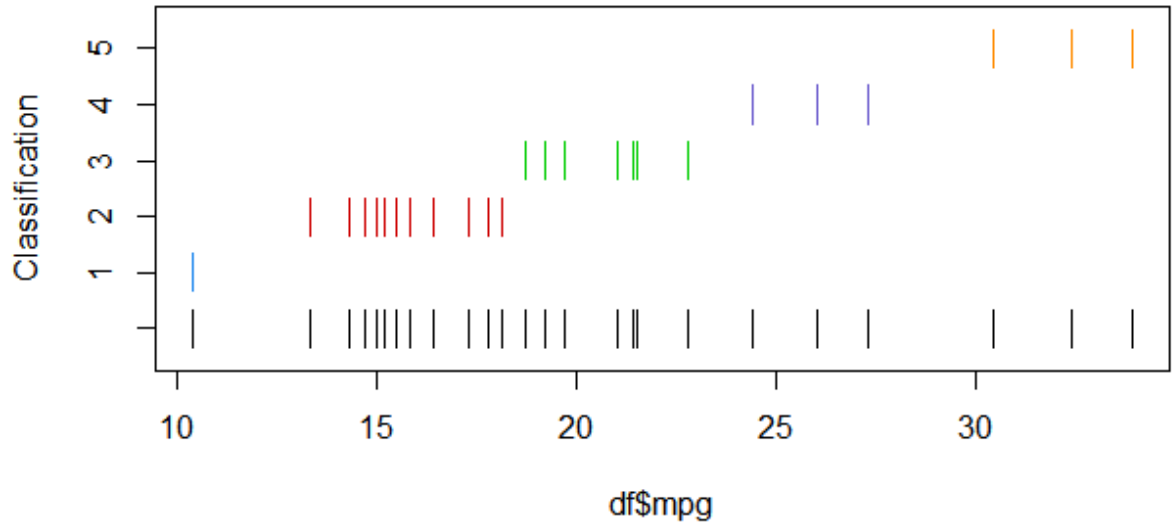

Рисунок 12

```
attributes(fit)

$names
 [1] "call"           "data"           "modelName"      "n"
 [5] "d"              "G"              "BIC"            "bic"
 [9] "loglik"         "df"             "hypvol"         "parameters"
[13] "z"              "classification" "uncertainty"

$class
[1] "Mclust"


summary(fit, parameter = TRUE)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust E (univariate, equal variance) model with 5 components:

 log-likelihood  n df       BIC       ICL
      -97.04211 32 10 -228.7416 -233.0352

Clustering table:
 1  2  3  4  5
 2 12 11  3  4

Mixing probabilities:
        1         2         3         4         5
0.06620433 0.35552096 0.35936612 0.09373896 0.12516963

Means:
       1        2        3        4        5
10.59454 15.68164 20.64789 25.80434 31.75727

Variances:
       1        2        3        4        5
2.099155 2.099155 2.099155 2.099155 2.099155
```

Вывод: Мы научились выполнять кластеризацию данных в R с помощью методов KMeans, иерархического, смеси гауссовских распределений.