МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8

по дисциплине «Языки программирования для статистической обработки данных»

Студент группы *ИМБО-11-23, Журавлев Ф.А.*

(подпись)

Преподаватель *Трушин СМ*

(подпись)

Москва 2025 г.

ЦЕЛЬ И ЗАДАЧИ

Цель практической работы:

Научиться проводить кластеризацию данных с использованием методов K-means и иерархической кластеризации в Python, R, а также визуализировать результаты кластерного анализа.

Задачи практической работы:

1. Выполнить кластеризацию методом K-means:
    - Python: использование библиотеки sklearn.
    - R: функция kmeans.
2. Провести иерархическую кластеризацию:
    - Python: библиотека scipy.cluster.hierarchy.
    - R: функции hclust, dendrogram.
3. Проанализировать результаты кластеризации:
    - Интерпретация кластеров (центроиды, количество объектов в каждом кластере).
    - Сравнение кластеров, полученных разными методами.
4. Визуализировать результаты кластеризации:
    - Python: графики кластеров с использованием matplotlib и seaborn.
    - R: графическое представление дендрограмм и кластеров.
5. Сравнить удобство выполнения кластерного анализа в Python, R.

РЕЗУЛЬТАТЫ ПРАКТИКИ

Шаг 1) Кластеризация в Python

1.1) K-means кластеризации в Python.

Сначала загрузим наши данные в питон, отберем только числовые признаки а также масштабируем данные:

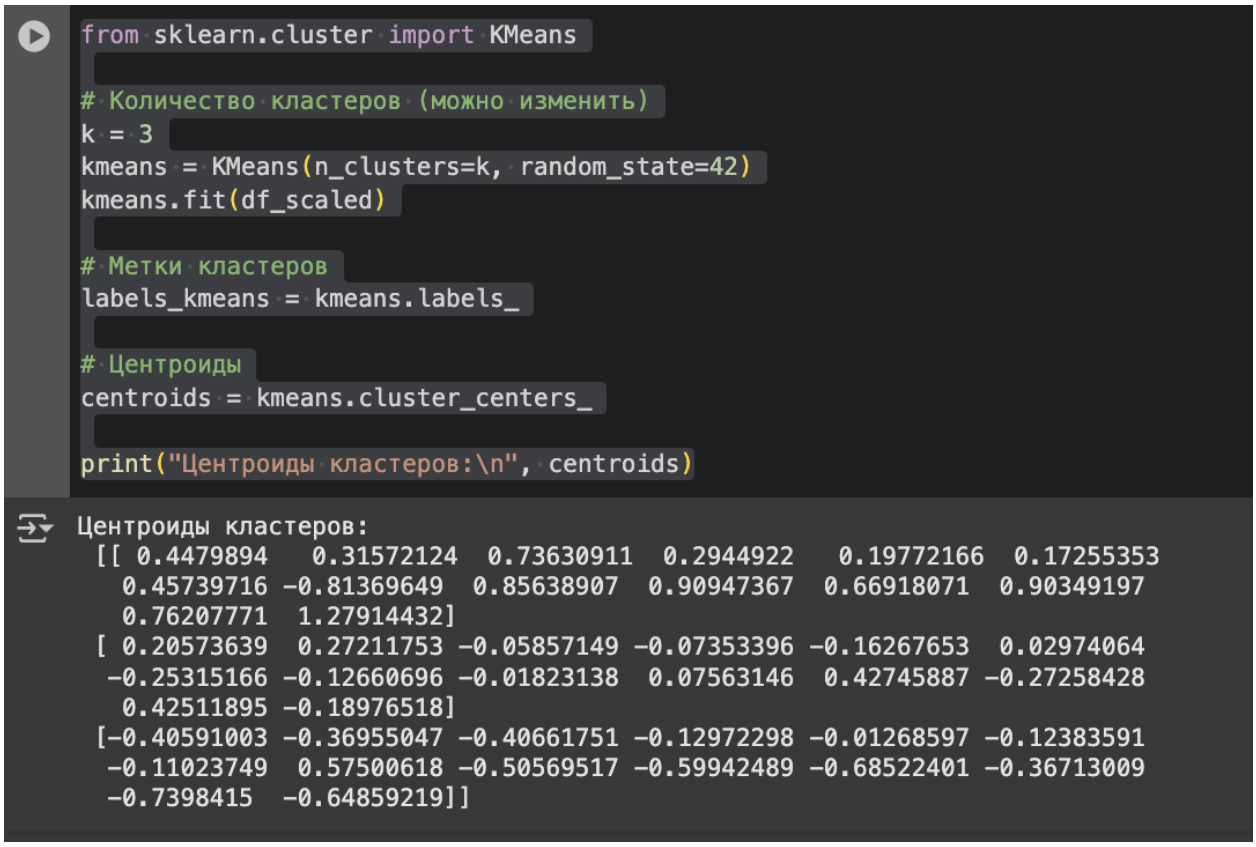
Рисунок 1.1 — Загрузка данных и обработка.

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt


# Выбираем только числовые признаки
df_numeric = df.select_dtypes(include='number')

# Масштабируем данные
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_numeric)
```

После загрузки исходной таблицы данных в формате .csv, следует провести кластеризацию двумя способами: K-means и иерархическую.

Напишем код, который реализует кластеризацию K-means и иерархическую, перед этим убедившись, что были добавлены все необходимые библиотеки:

Рисунок 1.2 — K-means кластеризация.



```
from sklearn.cluster import KMeans

# Количество кластеров (можно изменить)
k = 3
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(df_scaled)

# Метки кластеров
labels_kmeans = kmeans.labels_

# Центроиды
centroids = kmeans.cluster_centers_

print("Центроиды кластеров:\n", centroids)
```

```
Центроиды кластеров:
 [[ 0.4479894   0.31572124  0.73630911  0.2944922   0.19772166  0.17255353
   0.45739716 -0.81369649  0.85638907  0.90947367  0.66918071  0.90349197
   0.76207771  1.27914432]
 [ 0.20573639  0.27211753 -0.05857149 -0.07353396 -0.16267653  0.02974064
  -0.25315166 -0.12660696 -0.01823138  0.07563146  0.42745887 -0.27258428
   0.42511895 -0.18976518]
 [-0.40591003 -0.36955047 -0.40661751 -0.12972298 -0.01268597 -0.12383591
  -0.11023749  0.57500618 -0.50569517 -0.59942489 -0.68522401 -0.36713009
  -0.7398415  -0.64859219]]
```

Все результаты и итоги подведем в параграфе 3 «Сравнение результатов», а далее напишем код иерархической кластеризации.

Далее напишем код и рассмотрим график, который получился в результате K-means кластеризации указав, что количество кластеров равняется трем.

```python
from sklearn.decomposition import PCA

# Уменьшаем размерность до 2D
pca = PCA(n_components=2)
df_pca = pca.fit_transform(df_scaled)

# График кластеров
plt.figure(figsize=(8, 5))
plt.scatter(df_pca[:, 0], df_pca[:, 1], c=labels_kmeans, cmap='Set1', alpha=0.7)
plt.title("K-means кластеризация (PCA)")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.grid(True)
plt.show()
```

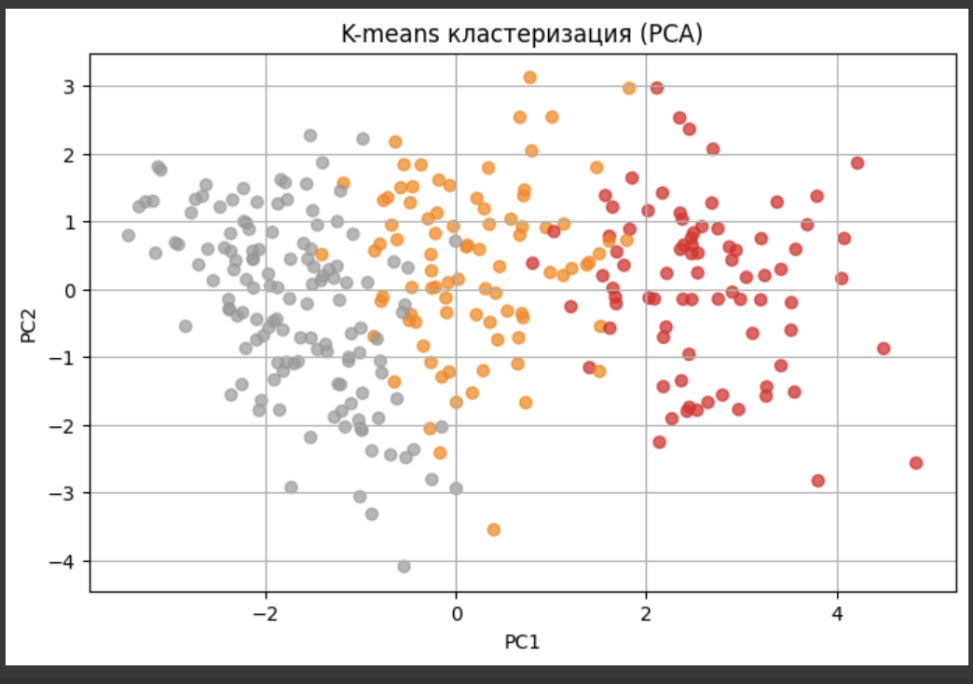


Рисунок 1.3 — Код кластеризации K-means.

1.2) Иерархическая кластеризация в Python.

```python
import scipy.cluster.hierarchy as sch
from scipy.cluster.hierarchy import fcluster

# Построение дендрограммы
Z = sch.linkage(df_scaled, method='ward')

plt.figure(figsize=(10, 5))
sch.dendrogram(Z)
plt.title("Иерархическая кластеризация — дендрограмма")
plt.xlabel("Объекты")
plt.ylabel("Расстояние")
plt.show()

# Присваиваем кластеры (например, 3 кластера)
labels_hier = fcluster(Z, t=3, criterion='maxclust')
```

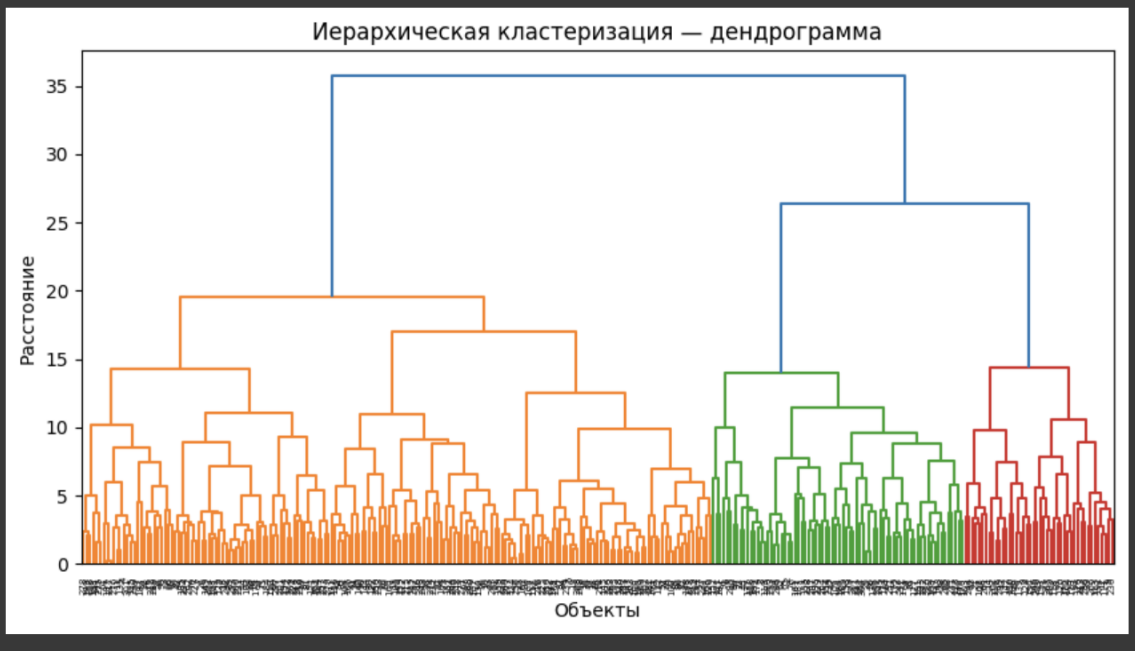
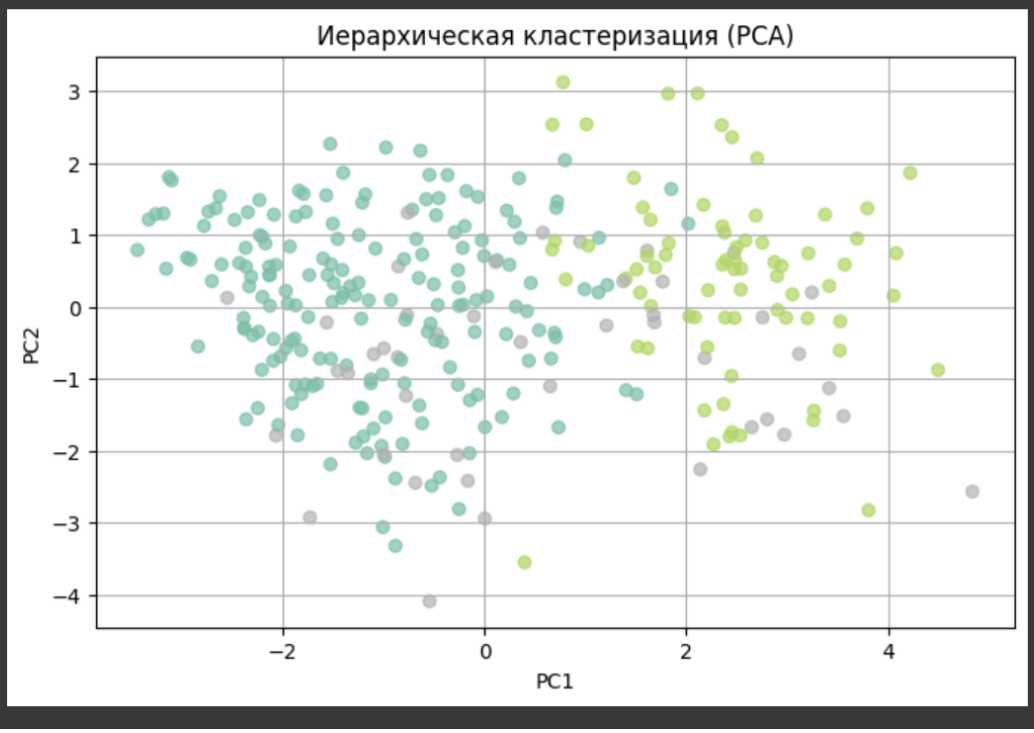


Рисунок 1.4 — Код иерархической кластеризации.

1.3) Визуализация иерархической кластеризация с PCA в Python.

```
plt.figure(figsize=(8, 5))
plt.scatter(df_pca[:, 0], df_pca[:, 1], c=labels_hier, cmap='Set2', alpha=0.7)
plt.title("Иерархическая кластеризация (PCA)")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.grid(True)
plt.show()
```



1.4) Анализ кластеров по количеству и средним в Python.

```python
import numpy as np

# Кол-во объектов в каждом кластере (K-means)
unique, counts = np.unique(labels_kmeans, return_counts=True)
print("Размеры кластеров (K-means):", dict(zip(unique, counts)))

# Средние значения по кластерам
df['cluster_kmeans'] = labels_kmeans
cluster_means = df.groupby('cluster_kmeans').mean()
print(cluster_means)
```

```
Размеры кластеров (K-means): {np.int32(0): np.int64(81), np.int32(1): np.int64(88), np.int32(2): np.int64(134)}
                      age       sex        cp    trestbps        chol  \
cluster_kmeans
0               58.481481  0.827160  3.864198  136.864198  256.913580
1               56.295455  0.806818  3.102273  130.397727  238.284091
2               50.776119  0.507463  2.768657  129.410448  246.037313

                     fbs   restecg     thalach     exang   oldpeak     slope  \
cluster_kmeans
0               0.209877  1.444444  131.024691  0.728395  2.093827  2.012346
1               0.159091  0.738636  146.715909  0.318182  1.127273  1.863636
2               0.104478  0.880597  162.738806  0.089552  0.344776  1.179104

                      ca      thal       num
cluster_kmeans
0               1.506173  6.197531  2.506173
1               0.409091  5.545455  0.704545
2               0.320896  3.291045  0.141791
```

Что значит каждый график будет расписано на 3 шаге

Шаг 2) Кластеризация в Rstudio.

2.1) Иерархическая и K-means кластеризация в Rstuido.

Далее проделаем все тоже самое, но уже с помощью языка программирования R, убедившись, что все необходимые пакеты были успешно установлены.

```r
set.seed(42)
k <- 3  # количество кластеров
kmeans_result <- kmeans(df_scaled, centers = k, nstart = 25)

# Метки кластеров
df$cluster_kmeans <- as.factor(kmeans_result$cluster)

# Кол-во объектов в каждом кластере
table(df$cluster_kmeans)

# Визуализация с PCA
fviz_cluster(kmeans_result, data = df_scaled,
             ellipse.type = "euclid",
             palette = "Set1",
             ggtheme = theme_minimal())
```

Рисунок 2.1 – код кластеризации K-means в R.

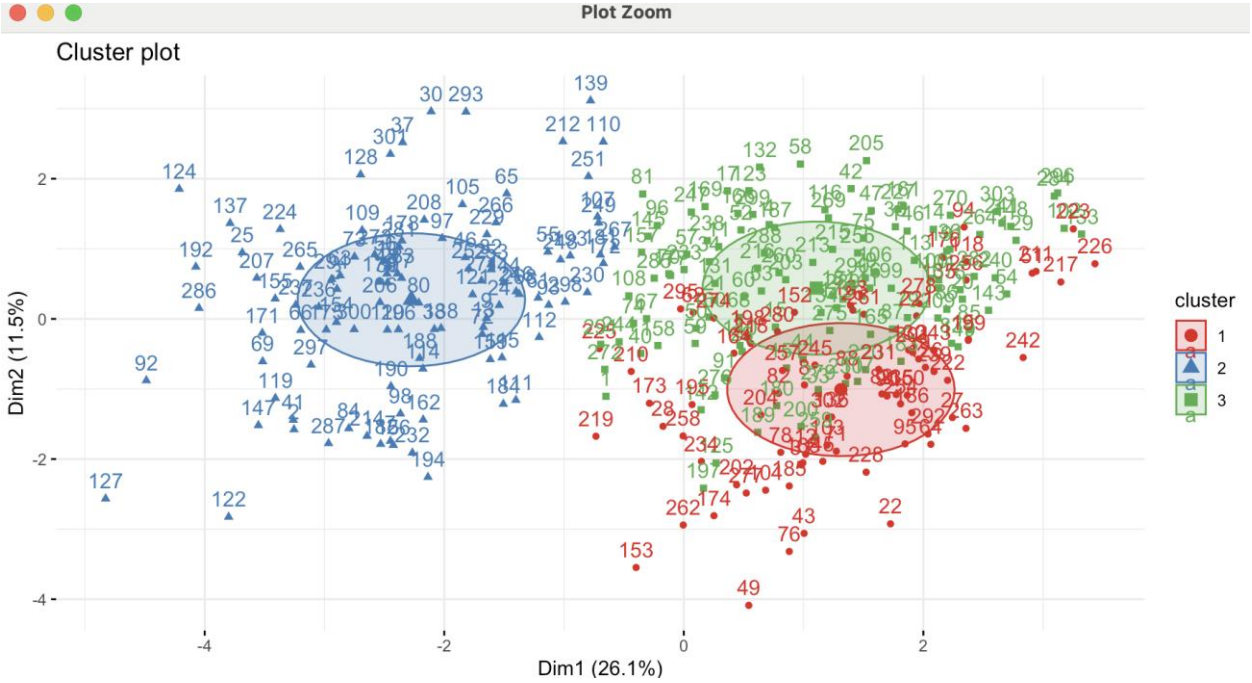


Рисунок 2.2 – график кластеризации K- means в R.

Построим иерархическую кластеризацию:

```r
# Расстояния
dist_matrix <- dist(df_scaled, method = "euclidean")

# Модель иерархической кластеризации
hc <- hclust(dist_matrix, method = "ward.D2")

# Построение дендрограммы
plot(hc, main = "Дендрограмма", xlab = "", sub = "")

# Разделим на 3 кластера
clusters_hier <- cutree(hc, k = 3)
df$cluster_hier <- as.factor(clusters_hier)
```

Рисунок 2.3 – код иерархической кластеризации в R.

И посмотрим на график:

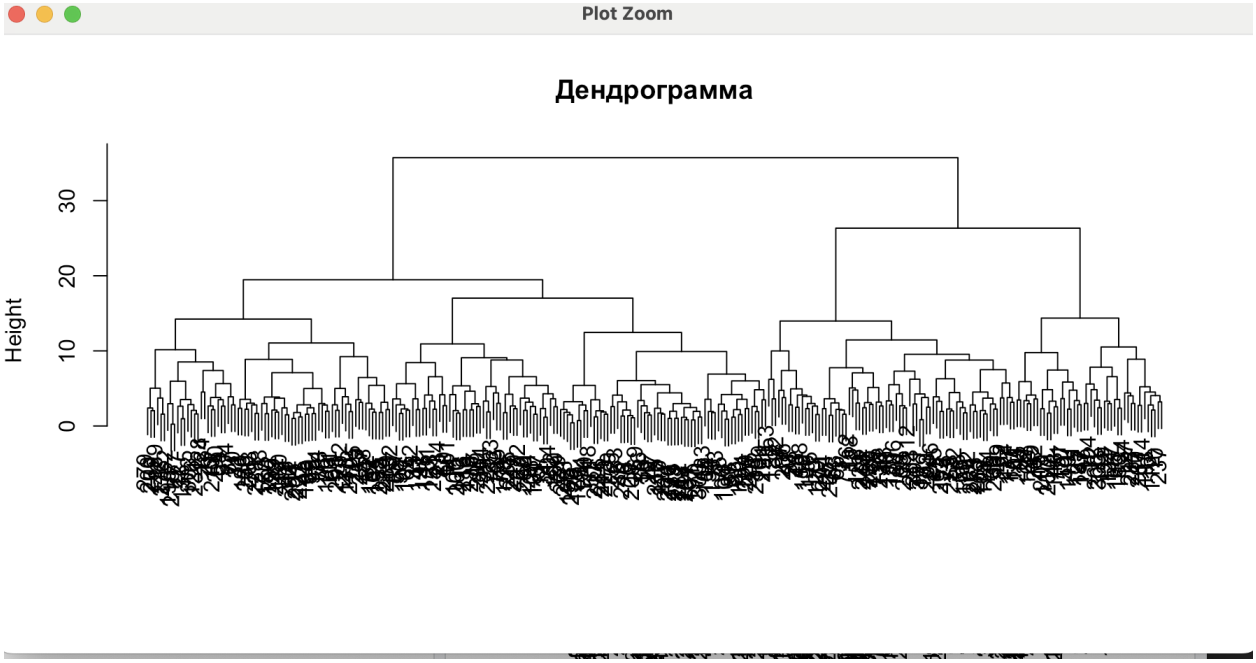


Рисунок 2.4 – График иерархической кластеризации в R.

Проведем визуализацию иерархических кластеров:

```r
fviz_cluster(list(data = df_scaled, cluster = clusters_hier),
             ellipse.type = "convex",
             palette = "Set2",
             ggtheme = theme_minimal())
```

Рисунок 2.5 – код визуализации иерархическиз кластеров в R.

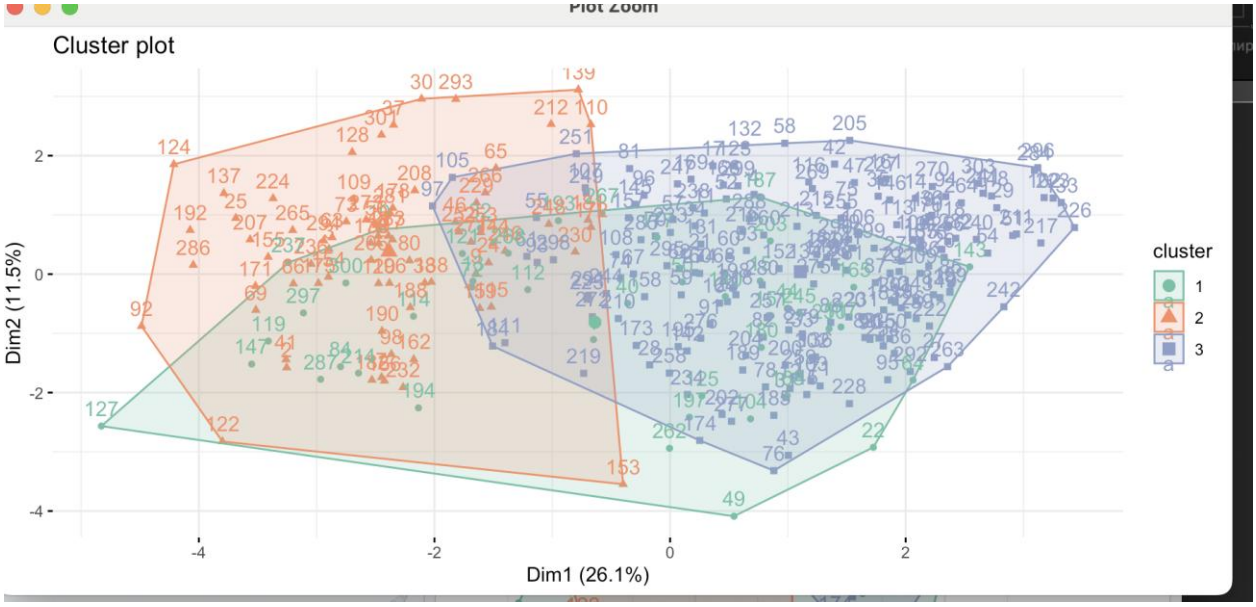


Рисунок 2.6 – График визуализации иерархических кластеров в R.

Проведем анализ кластеров:

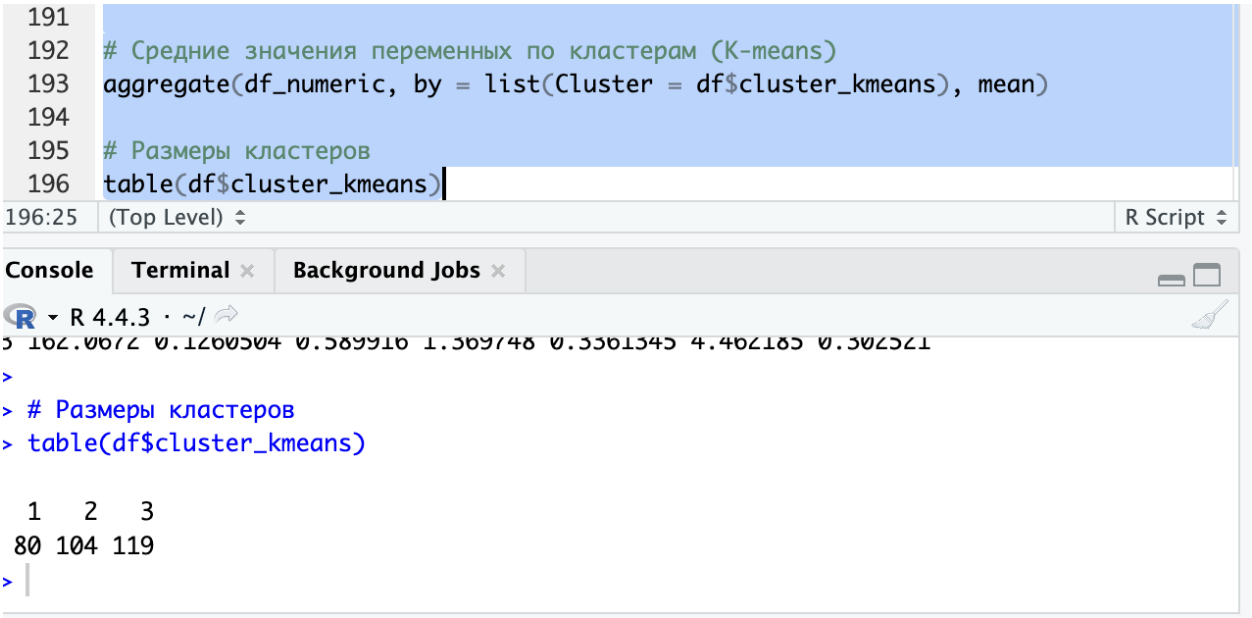


Рисунок 2.7 – Анализ кластеров.

3 СРАВНЕНИЕ РЕЗУЛЬТАТОВ

В ходе практической работы были применены два метода кластеризации: **K-means** и **иерархическая кластеризация**. Анализ проводился в средах Python и R. Ниже представлены основные различия и особенности каждого подхода, а также описание всех графиков, полученных в ходе работы.

Сравнение методов

- **K-means** — это метод, при котором изначально задаётся количество кластеров (например, 3), и алгоритм ищет оптимальное разбиение данных на группы, минимизируя внутрикластерные расстояния. Он работает быстро и хорошо подходит для больших объёмов данных.
- **Иерархическая кластеризация** строит древовидную структуру объединения объектов, начиная с отдельных точек. В отличие от K-means, количество кластеров можно выбрать позже, визуально — на основе дендрограммы.

Интерпретация графиков

1. **Диаграмма кластеров K-means на основе PCA**

   Этот график показывает, как объекты (наблюдения) были разбиты на кластеры методом K-means. Каждая точка — это один объект из выборки. Цвет отражает принадлежность к определённому кластеру. Использование PCA позволяет изобразить многомерные данные на плоскости в виде двух главных компонент, сохранив основную структуру. Если кластеры визуально хорошо разделяются, это означает, что алгоритм нашёл естественные группы в данных.

2. **Дендрограмма (иерархическая кластеризация)**

Этот график показывает процесс последовательного объединения объектов в кластеры. По оси Y откладывается расстояние между объединяемыми группами. Чем выше соединяются ветви — тем менее похожи объединяемые группы. Чтобы выбрать, например, 3 кластера, можно мысленно провести горизонтальную линию, пересекающую три ветви. Дендрограмма удобна тем, что позволяет увидеть структуру данных даже до выбора количества кластеров.

3. **Диаграмма кластеров после иерархической кластеризации (PCA)**

   Аналогична графику K-means, но отражает результат иерархического подхода. Отображает, как алгоритм разделил данные на кластеры после «разреза» дендрограммы. График позволяет сравнить полученные группы с теми, что дал K-means: совпадают ли кластеры, есть ли перемешанные точки и т.д.

4. **Таблица средних значений по кластерам**

   Эта таблица содержит усреднённые значения всех признаков по каждому кластеру. Она позволяет понять, чем именно различаются группы: например, один кластер может объединять пациентов с высоким пульсом, другой — с повышенным давлением. Такая интерпретация делает результат кластеризации понятным с точки зрения предметной области (в данном случае — медицина).

Общий вывод

Оба метода — K-means и иерархическая кластеризация — выделили группы в данных. При этом визуализация показала, что результаты

кластеризации в целом согласуются: группы перекрываются частично, но в целом отражают структуру данных.

Метод K-means удобен при чётко заданном количестве кластеров, тогда как иерархическая кластеризация более гибкая и даёт лучшее представление о вложенности и близости объектов. Оба подхода успешно применимы к медицинским данным, подобным heart_cleaned.csv.

ВЫВОДЫ

В результате выполнения практической работы мы ознакомились с методами кластеризации данных, включая **K-means** и **иерархическую кластеризацию**. Были изучены основные этапы применения этих алгоритмов, выполнено масштабирование данных, визуализация кластеров и интерпретация полученных результатов. Также была проведена работа в двух средах программирования — **Python и R**, что позволило сравнить их удобство и функциональные возможности при анализе данных.

В завершение были сделаны общие выводы о применимости кластерного анализа к медицинским данным и его роли в выявлении скрытых групп и закономерностей в выборке.